

Faculdade de Engenharia da Universidade do Porto



Sistema de Controlo Pró-Activo num Contexto Habitacional

Ricardo Jorge dos Santos Garrido

Dissertação realizada no âmbito do Mestrado Integrado em Engenharia
Electrotécnica e de Computadores
Major Telecomunicações

Orientador: Prof. Dr. Ricardo Morla

Co-orientador: Eng. Filipe Sousa

Junho de 2008

A Dissertação intitulada

“Sistema Multimédia e de Controlo Pró-Activo num Contexto Habitacional”

foi aprovada em provas realizadas 10/Julho/2008

o júri

Presidente Professor Doutor Pedro Henrique Henriques Guedes de Oliveira
Professor Catedrático da Faculdade de Engenharia da Universidade do Porto

Professor Doutor Carlos Manuel Robalo Lisboa Bento
Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Professor Doutor Ricardo Santos Morla
Professor Auxiliar Convidado da Faculdade de Engenharia da Universidade do Porto

O autor declara que a presente dissertação (ou relatório de projecto) é da sua exclusiva autoria e foi escrita sem qualquer apoio externo não explicitamente autorizado. Os resultados, ideias, parágrafos, ou outros extractos tomados de ou inspirados em trabalhos de outros autores, e demais referências bibliográficas usadas, são correctamente citados.

Autor - Ricardo Jorge dos Santos Garrido

Faculdade de Engenharia da Universidade do Porto

Resumo

Nos dias de hoje a tecnologia está cada vez mais presente nas nossas habitações, desde os electrodomésticos com acção programável aos sistemas automáticos de iluminação e climatização. Estes equipamentos possuem os seus próprios sensores e actuam individualmente, o que geralmente não representa a melhor solução do ponto de vista global pois eles podem reagir a acções de outros equipamentos dando origem a um comportamento instável em que a situação de equilíbrio, se atingida, não será a ideal em termos de eficiência energética. Torna-se necessário desenvolver um agente racional que faça a gestão de redes ambiente de forma a torná-las mais eficientes. A capacidade das redes neuronais na classificação e previsão de padrões é reconhecida, assim como o seu poder de generalização. A aplicação deste tipo de sistemas computacionais no desenvolvimento de ambientes habitacionais inteligentes não foi ainda completamente testada. A inteligência e racionalidade destes agentes torna possível, através da observação das actividades dos habitantes, prever e automatizar algumas das suas tarefas. É possível também relacionar informação obtida de vários dispositivos e assim detectar possíveis situações de emergência, actuando, de forma autónoma, para garantir a segurança dos ocupantes e da habitação. Para além de um módulo de inteligência responsável pela gestão dos equipamentos foi também desenvolvida uma aplicação que permite fazer a análise de vários parâmetros dos nós da rede neuronal. Desta forma tornou-se mais imediato perceber a influência das alterações realizadas no comportamento da rede. Como não era viável o desenvolvimento desta solução num ambiente físico, foi necessário implementar um simulador de habitação, denominado HomeSim, para se poder reproduzir as actividades dos habitantes. A interligação com a rede IP permite-nos controlar os dispositivos da habitação a partir de um interface *web* e receber notificações sobre eventos pré-definidos como a presença de pessoas na habitação ou um pedido de ajuda.

Abstract

Nowadays technology is ever present in our homes, from programmable appliances to automated lighting and heating systems. These devices have their own sensors and they work individually, which isn't always the best global solution because they can react to each other's actions leading to an unstable behavior in which a balance is never achieved in the most energy-efficient way. It's necessary to develop a rational agent that can manage ambient networks, making them more efficient. The ability of neural networks to classify and predict patterns is well known, as is their power of generalization. The use of this type of computational systems in the development of intelligent home environments has not yet been fully tested. This agent's intelligence and rationality allows it, by the observation of the inhabitant's activities, to predict and automate some of their tasks. It's also possible to relate information collected from different devices and use it to detect possible emergency situations, acting, in an autonomous way, to ensure the safety of the inhabitants and their home. Besides an intelligence module responsible for the management of the home's equipments, an application that allows analyzing several parameters of the network's nodes has also been developed. With it the influence of the changes made in the network's behavior became clearer. Because it was not possible to develop the intelligence module in a real environment it was necessary to create a home simulator, the HomeSim, to replicate the inhabitant's activities. The interconnection with the IP network allows us to control our home's devices from any web interface and to receive notifications over predefined events, like the presence of people in the house or a call for help.

Agradecimentos

Quero agradecer à minha família, à minha namorada e a todos aqueles que me apoiaram ao longo do curso. Agradeço ao Professor Doutor Ricardo Morla e ao Eng. Filipe Sousa pela constante disponibilidade, crítica construtiva e esclarecimentos dados e ao INESC por me ter proporcionado os meios necessários para o desenvolvimento desta dissertação.

O autor

“Find a job you love and you'll never work a day in your life”

Confucious

Índice

| | | |
|-------|---|----|
| 1 | Introdução | |
| 1.1 | Motivação | 24 |
| 1.2 | Descrição do Problema | 24 |
| 1.3 | Objectivos do trabalho | 25 |
| 1.4 | Estrutura da dissertação | 25 |
| 2 | Revisão de Tecnologias Utilizadas | |
| 2.1 | Redes sem fios | 27 |
| 2.1.1 | Tipos fundamentais de dispositivos ZigBee | 30 |
| 2.1.2 | Perfis de dispositivos ZigBee | 31 |
| 2.1.3 | Segurança na comunicação ZigBee | 31 |
| 2.2 | Web services | 32 |
| 2.2.1 | Principais tecnologias utilizadas pelos <i>web services</i> | 32 |
| 2.3 | .Net Framework | 33 |
| 2.3.1 | Windows Workflow Foundation | 34 |
| 2.3.2 | Windows Presentation Foundation | 34 |
| 2.3.3 | Windows Communication Foundation | 35 |
| 2.4 | Redes neuronais | 36 |
| 2.4.1 | Arquitectura das redes | 39 |
| 2.4.2 | Capacidade de representação | 40 |

| | | |
|-------|---|----|
| 2.4.3 | Treino das redes neurais | 40 |
| 2.4.4 | Rede de retropropagação | 41 |
| 2.4.5 | Regra delta generalizada..... | 42 |
| 2.4.6 | Passo de aprendizagem..... | 44 |
| 2.4.7 | Critério de paragem..... | 44 |
| 2.4.8 | Momento | 45 |
| 3 | Trabalho Relacionado | |
| 3.1 | MavHome..... | 47 |
| 3.1.1 | Algoritmos de previsão | 49 |
| 3.1.2 | Experiências e testes | 49 |
| 3.2 | Amigo 50 | |
| 3.3 | Resumo | 52 |
| 4 | Arquitectura do Sistema | |
| 4.1 | Arquitectura | 53 |
| 4.2 | Requisitos do Sistema | 54 |
| 5 | Implementação Rede Neuronal HomeSim | |
| 5.1 | Arquitectura da rede | 59 |
| 5.1.1 | Escolha do número de camadas | 60 |
| 5.1.2 | Escolha do número de nós das várias camadas | 60 |
| 5.1.3 | Treino da rede | 63 |
| 5.1.4 | Pârametros de ajuste | 63 |
| 5.2 | Implementação..... | 64 |
| 5.2.1 | A plataforma NeuronDotNet | 65 |
| 5.2.2 | Implementação das rede neuronais | 65 |
| 5.3 | Rede Neuronal Visual..... | 65 |
| 5.4 | Resumo | 66 |
| 6 | Implementação Aplicação HomeSim | |

| | | |
|-------|---|----|
| 6.1 | Implementação | 67 |
| 6.1.1 | Codificação dos valores de entrada..... | 69 |
| 6.1.2 | Codificação dos valores de saída | 73 |
| 6.2 | Resumo | 75 |
| 7 | Avaliação do Trabalho | |
| 7.1 | Cenário de teste..... | 77 |
| 7.2 | Testes realizados..... | 78 |
| 7.2.1 | Teste da rede com 31 nós escondidos | 78 |
| 7.2.2 | Teste da rede com 13 nós escondidos..... | 79 |
| 7.2.3 | Teste com alteração de parâmetros da rede..... | 80 |
| 7.2.4 | Teste do número de intervenções do utilizador | 81 |
| 7.3 | Análise de resultados..... | 82 |
| 8 | Conclusões e trabalho Futuro | |
| 8.1 | Satisfação dos objectivos | 85 |
| 8.2 | Revisão do trabalho desenvolvido | 86 |
| 8.3 | Trabalho futuro..... | 86 |

Lista de Figuras

| | |
|--|----|
| FIGURA 2.1 - CONDIÇÕES DE APLICAÇÃO DAS TECNOLOGIAS DE REDE SEM FIOS MAIS UTILIZADAS. | 28 |
| FIGURA 3.1 - ARQUITECTURA COMPLETA DO MAVHOME. | 48 |
| FIGURA 3.2 - ARQUITECTURA COMPLETA DO PROJECTO AMIGO. | 51 |
| FIGURA 4.1 - RELAÇÃO ENTRE AS DIFERENTES APLICAÇÕES. | 54 |
| FIGURA 5.1 - FUNÇÕES IMPLEMENTADAS PELA APLICAÇÃO REDE NEURONAL HOMESIM. | 64 |
| FIGURA 5.2 - INTERFACE DA APLICAÇÃO REDE NEURONAL VISUAL. | 66 |
| FIGURA 6.1 - FUNÇÕES IMPLEMENTADAS PELO HOMESIM. | 68 |
| FIGURA 7.1 - TESTE DA REDE COM 31 NÓS ESCONDIDOS. | 79 |
| FIGURA 7.2 - TESTE DA REDE COM 13 NÓS ESCONDIDOS. | 80 |
| FIGURA 7.3 - TESTE DA REDE COM 13 NÓS ESCONDIDOS, PASSO DE APRENDIZAGEM 0.2 E MOMENTO 0.03. | 81 |
| FIGURA 7.4 - NÚMERO DE INTERVENÇÕES DO HABITANTE DEVIDO A UM ERRO DE PREVISÃO. | 82 |
| <u>ANEXO</u> - FIGURA 1 - MENU DE ESCOLHA DE PERSONAGEM. | 89 |
| <u>ANEXO</u> - FIGURA 2 - SALA DE ESTAR. | 90 |
| <u>ANEXO</u> - FIGURA 3 - ESCRITÓRIO. | 90 |
| <u>ANEXO</u> - FIGURA 4 - QUARTO. | 91 |
| <u>ANEXO</u> - FIGURA 5 - QUARTO DE BANHO. | 91 |

Lista de Tabelas

| | |
|---|----|
| TABELA 2.1 - COMPARAÇÃO ENTRE AS VÁRIAS TECNOLOGIAS. | 28 |
| TABELA 4.1 - CENÁRIO A: APRENDIZAGEM DE UMA ROTINA DO UTILIZADOR A. | 55 |
| TABELA 4.2 - CENÁRIO B: INTERPRETAÇÃO DO SISTEMA QUANDO CONFRONTADO COM UM NOVO CENÁRIO. | 56 |
| TABELA 6.1 - CODIFICAÇÃO DAS HORAS EM VALORES ACEITES POR UMA REDE NEURONAL. | 70 |
| TABELA 6.2 - CODIFICAÇÃO DOS MINUTOS. | 71 |
| TABELA 6.3 - CODIFICAÇÃO DO NÚMERO DE IDENTIFICAÇÃO DOS EQUIPAMENTOS DA HABITAÇÃO. | 72 |
| TABELA 6.4 - CODIFICAÇÃO DOS VALORES DE TEMPERATURA. | 73 |
| TABELA 6.5 - CODIFICAÇÃO DA INTENSIDADE LUMINOSA DE SAÍDA. | 74 |
| TABELA 6.6 - CODIFICAÇÃO DA TEMPERATURA DE SAÍDA. | 74 |
| TABELA 6.7 - CODIFICAÇÃO DA BANDA/CANAL DE SAÍDA. | 75 |
| TABELA 7.1 - CENÁRIO DE TESTE. | 78 |

Definições, Abreviaturas e Símbolos

Definições:

Coordenador Zigbee- Um dispositivo IEEE 802.15.4, FFD, responsável pela formação de uma rede Zigbee, e pela gestão das ligações dos dispositivos associados.

Endereço de *Endpoint* - Endereço atribuído a cada *endpoint*, para além do endereço único IEEE de 64 bits atribuído ao dispositivo e do endereço de 16 bits atribuído à rede a que pertence.

Endpoint - Um componente controlado por um dispositivo com interface Zigbee. Cada dispositivo Zigbee pode controlar até 240 componentes.

FFD (*Full Function Device*) - Dispositivo com uma boa capacidade de processamento e de memória, que é capaz de formar redes com capacidade de ligação ponto-a-ponto para interligação de dispositivos RFD.

Jitter - É uma variação estatística do atraso na entrega de dados numa rede, ou seja, pode ser definido como a medida de variação do atraso entre os pacotes sucessivos de dados.

Rede emalhada - Uma rede em que o roteamento de mensagens é realizado de forma descentralizada e cooperativa, na qual as comunicações entre duas unidades pode ser repetida sucessivamente pelas unidades existentes na rede até ao destinatário.

RFD (*Reduced Function Device*) - Dispositivo com poder de processamento e capacidade de memória limitados. Apenas comunicam com FFDs.

Router ZigBee - Dispositivo IEEE 802.15.4 que não actua como coordenador de rede Zigbee mas que desempenha uma função semelhante dentro do seu espaço operacional. É capaz de reencaminhar mensagens entre dispositivos e suporta associações.

Serviço - Funcionalidade oferecida para a qual um interface, ponto de acesso, funcionalidade e qualidade de serviço estão especificados.

SOA (Service-Oriented Architecture) - É uma colecção de serviços que utilizam a troca de mensagens como forma de comunicação. A comunicação pode ser entre dois serviços ou pode ser a coordenação de uma actividade por parte de vários serviços.

WPAN (*Wireless Personal Area Network*) - São áreas de pequena dimensão onde se enquadram tecnologias wireless de pequeno alcance, entre 10 e 100 metros. Esta área compreende as redes sem fios que utilizam dispositivos como PDAs, PCs ou periféricos, sendo o campo de tecnologias como o Bluetooth (responsabilidade do subgrupo IEEE 802.15.1) ou IrDA. O Low Rate WPAN, que reúne características tais como uma baixa *data rate*, baixa complexidade e um reduzido consumo energético, faz parte integrante deste campo e é sobre ele que foi desenvolvido o ZigBee, conjunto de especificações da responsabilidade do subgrupo de trabalho IEEE 802.15.4.

Lista de Abreviaturas:

| | |
|------|--|
| FBR | Função de Base Radial |
| MSV | Máquinas de Vector de Suporte |
| RNA | Rede Neuronal Artificial |
| RNH | Rede Neuronal HomeSim |
| RNV | Rede Neuronal Visual |
| SOAP | Simple Object Access Protocol |
| XAML | Extensible Application Markup Language |
| XML | Extensible Markup Language |
| WSDL | Web Service Definition Language |

Lista de Símbolos:

| | |
|----------|---|
| α | Momento |
| η | Passo de aprendizagem |
| e_k | Erro entre o valor de saída obtido e o valor de saída desejado da rede neuronal |
| w | Peso de uma ligação da rede neuronal |
| y_k | Valor de saída desejado de uma rede neuronal |
| z_k | Valor de saída obtido de uma rede neuronal |

Capítulo 1

Introdução

Este trabalho surge no seguimento do projecto de fim de curso em que foi desenvolvido um sistema que permite o controlo de equipamentos de uma habitação, interligados por uma rede ZigBee, através de dispositivos móveis de uso generalizado como telemóveis e PDAs, assim como a partir de quaisquer dispositivos com suporte IP ou interface *web*.

A inteligência em redes ambiente tem como foco a utilização da tecnologia na criação de ambientes electrónicos que são sensíveis e reactivos à presença de pessoas, por forma a tornar a sua vida mais simples e confortável. Os elementos-chave destes ambientes são a ubiquidade, a inteligência e a acção natural e transparente. A ubiquidade representa a situação em que estamos rodeados por inúmeros sistemas embebidos interligados que servem o utilizador mas são, quanto possível, invisíveis a este. A inteligência advém do envolvimento digital ser capaz de analisar o contexto, adaptar-se às pessoas que nele vivem e aprender o seu comportamento. Tirando partido desta aprendizagem, o sistema deve actuar de forma natural e transparente ao utilizador, tendo como prioridade o bem-estar das pessoas.

1.1 Motivação

Hoje em dia, e cada vez mais, existem equipamentos e dispositivos nas nossas casas que melhoram a nossa qualidade de vida. Os electrodomésticos inteligentes e os sistemas de controlo e iluminação são bons exemplos de equipamentos que aumentam o conforto e que trouxeram maior poder e possibilidades de personalização às nossas vidas. Porém eles necessitam, em certa medida, de interacção humana para poderem funcionar de acordo com as nossas pretensões, actuando de forma independente para satisfazer as preferências dos utilizadores. Este comportamento não permite atingir a melhor solução global em termos de eficiência energética. Existe a necessidade de relacionar factores como as condições ambientais e o consumo energético dos equipamentos cuja acção pode ser utilizada para cumprir um mesmo propósito, gerindo-os de forma adequada tendo sempre como objectivo o bem-estar e segurança dos habitantes. Pretende-se também analisar a validade da utilização de redes neuronais na implementação de sistemas habitacionais inteligentes.

1.2 Descrição do Problema

As habitações são áreas muito pessoais. Uma solução implementada para criar uma casa inteligente não pode ser intrusiva, devendo estar num plano secundário pois apesar de acrescentar funcionalidade não deve nunca exercer um controlo excessivo. A questão importante é como saber quando um sistema deixa de actuar de forma correcta, transparente e discreta, e passa a actuar de forma intrusiva e abusiva. Esse limiar é subjectivo e depende também do julgamento das pessoas que estão envolvidas. O campo da inteligência artificial não está ainda desenvolvido ao ponto de conseguir identificar onde se situa essa fronteira pessoal, os níveis de aceitação de um ser humano. A solução passa pelo sistema tomar uma postura retraída, contida, até ser decidido, com um elevado nível de confiança, que uma acção deve ser tomada. Este é o campo em que a inteligência artificial aplicada a habitações se movimenta, aquele em que dez decisões acertadas não compensam uma decisão errada levando a uma recusa do sistema, que dessa forma não pode evoluir até níveis de eficácia superiores.

A eficiência energética é uma preocupação crescente nos dias de hoje. As soluções com grande foco na redução do consumo de energia são ainda encaradas, frequentemente, de forma negativa por serem interpretadas como sistemas que nos obrigam a abdicar de algumas das nossas preferências. Em suma, as pessoas desejam poupar energia, mas não à custa do seu bem-estar.

1.3 Objectivos do trabalho

Um dos objectivos deste trabalho é ajudar a perceber como é possível transformar o ambiente habitacional num agente inteligente que actue da mesma forma que um ser humano, quando confrontado com a mesma situação e condições. Essa tarefa passa por criar um módulo de inteligência que deve ser capaz de aprender e progressivamente automatizar as actividades recriadas pelos habitantes, quando forem detectados os padrões de comportamento associados a rotinas. Pretende-se satisfazer as preferências dos utilizadores, reduzindo, sempre que possível, o consumo de energia. Um dos pontos chave deste problema é fazer com que os equipamentos deixem de actuar de forma individual, pela avaliação de factores singulares, para passarem a ser geridos e controlados por um agente que faz uma avaliação inteligente de um conjunto de factores importantes para o seu funcionamento, dum ponto de vista global. Este agente dispõe desses mesmos equipamentos e controla-os, de forma eficiente, para que os interesses dos ocupantes sejam atingidos, com o acréscimo de serem tomadas as opções que maximizam a eficiência energética.

1.4 Estrutura da dissertação

Este trabalho encontra-se estruturado em oito capítulos dos quais o primeiro é composto por esta introdução ao trabalho.

No segundo capítulo procede-se ao estudo do estado da arte das tecnologias utilizadas na implementação deste trabalho.

No terceiro capítulo é feita a análise dos projectos previamente realizados na área da aprendizagem de máquina aplicada a ambientes habitacionais.

No quarto capítulo descreve-se a arquitectura e estrutura do sistema implementado e faz-se a análise aos requisitos necessários e decisões tomadas durante o seu desenvolvimento.

No quinto capítulo indicam-se as várias etapas da implementação da Rede Neuronal HomeSim, a aplicação que actua como o agente racional da habitação.

No sexto capítulo descreve-se o desenvolvimento do HomeSim, um simulador de habitação que permite recriar algumas das actividades comuns num contexto habitacional.

No sétimo capítulo procede-se à avaliação do trabalho realizado, através do seu desempenho em cenários estabelecidos.

No oitavo capítulo são discutidas as conclusões gerais deste trabalho e analisados os resultados obtidos, sendo igualmente apresentadas algumas perspectivas de desenvolvimento futuro.

Capítulo 2

Revisão de Tecnologias Utilizadas

Neste capítulo faz-se a análise das tecnologias consideradas para a implementação deste projecto, com especial atenção às que foram escolhidas. São analisadas as vantagens e desvantagens de cada uma.

2.1 Redes sem fios

A estrutura de uma habitação favorece a utilização de redes sem fios pois a utilização de uma rede cablada, apesar de poder ser montada sem grandes inconvenientes durante o processo de construção da casa, tem grandes desvantagens. A mobilidade, fácil manutenção e a simples escalabilidade do número de equipamentos são características adequadas ao contexto habitacional. Das várias tecnologias existentes com estas características, as mais interessantes são o Wi-Fi, o Bluetooth e o ZigBee. A tecnologia Wi-Fi é já nos dias de hoje largamente utilizada em empresas e lares e tem sido alvo de grande evolução nos últimos tempos, suportando grandes débitos que permitem, inclusivé, a visualização de vídeo de alta definição. Na gama de tecnologias de redes sem fios com menores débitos e consumos, a mais implantada é o Bluetooth, estando disponível na maioria dos telemóveis e outros equipamentos portáteis. Na sua especificação estão previstos diferentes perfis que incluem funcionalidades que vão desde a transmissão de áudio até à formação de redes IP, com débitos que atingem os 2.1 Mbit/s. Todavia a sua utilização é desaconselhada, uma vez que os dispositivos não podem entrar em modo de poupança de energia sem que a ligação seja quebrada, requerendo nesta condição uma reorganização da rede e consequente aumento de consumo. Uma variante do actual Bluetooth, o Wibree, está agora a ser desenvolvida e permitirá obter uma poupança substancial no consumo energético, nomeadamente através do modo adormecido, que não obriga a uma quebra de ligação.

| Tecnologia Padrão | ZigBee™ 802.15.4 | Wi-Fi™ 802.11b | Bluetooth™ 802.15.1 |
|---------------------------------------|-----------------------------|----------------------------------|--------------------------|
| Foco | Monitorização e controlo | Internet, vídeo e e- mail. | Substituição de cabos |
| Recursos do sistema | 4KB-32KB | 1MB+ | 250KB+ |
| Autonomia da bateria (dias) | 100-1000+ | 0.5-5 | 1-7 |
| Dimensão da rede | 2 ⁶⁴ | 32 | 7 |
| Largura de banda | 20-250 | 11.000+ | 720 |
| Alcance da comunicação (metros) | 1-100+ | 1-100 | 1-10+ |
| Métricas de sucesso | Fiabilidade Custo | Velocidade Flexibilidade | Custo Conveniência |

Tabela 2.1 - Comparação entre as várias tecnologias.

As novas tecnologias sem fios para redes de baixo débito e baixo consumo estão agora em expansão. Cada tecnologia de rede sem fios foi concebida para atender a determinados requisitos de débito e alcance de comunicação, como se pode ver na Figura 2.1.

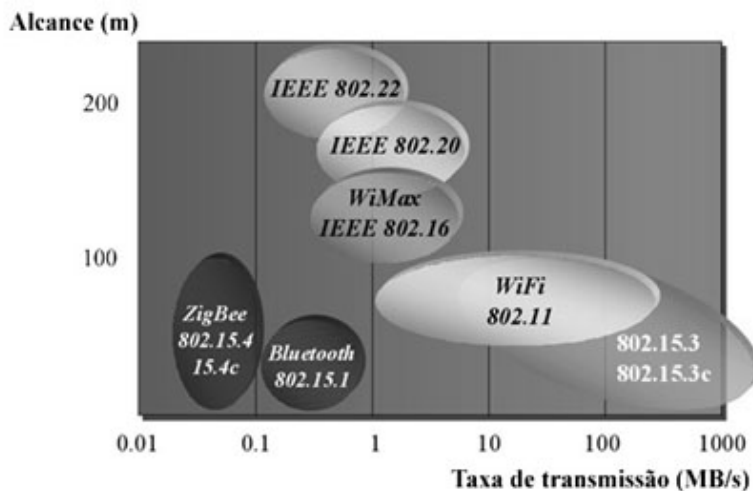


Figura 2.1 - Condições de aplicação das tecnologias de rede sem fios mais utilizadas.

O ZigBee é a nova tecnologia emergente vocacionada para a domótica, assente num protocolo aberto.

As principais características do ZigBee são:

- *Duty-cycle* baixo (< 0,1%);
- Capacidade do dispositivo de permanecer inactivo durante um longo período de tempo, permitindo a um *end device* uma esperança de vida de vários meses ou mesmo anos;
- Elevada densidade de nós por rede, sendo assim indicada para redes de sensores;
- Espaço de endereçamento para 18,450,000,000,000,000 dispositivos (endereço de 64 bits IEEE);
- Opção de *time slot* garantido para aplicações que necessitam de uma baixa latência;
- Acesso ao canal do tipo CSMA-CA, que permite um débito elevado e uma latência baixa para dispositivos com um baixo *duty-cycle* como sensores ou controladores;
- Suporte de topologias estáticas e dinâmicas, tais como estrela, árvore e emalhada.

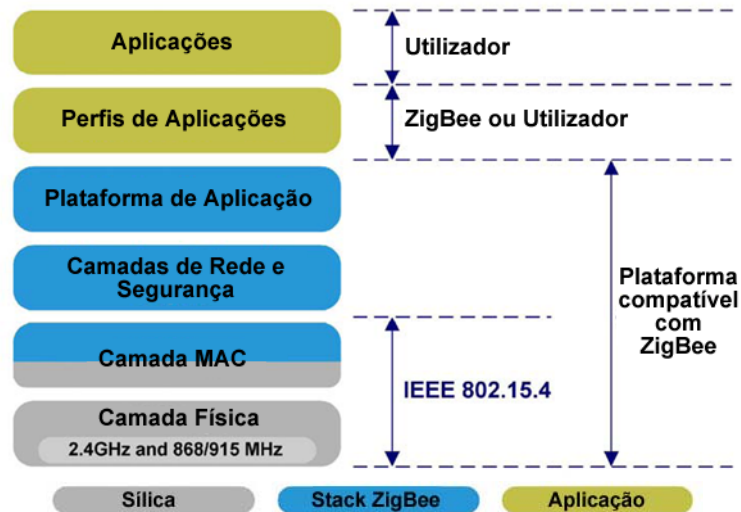


Figura 2.2 - Estrutura em camadas do IEEE 802.15.4 e do Zigbee.

As comunicações por ZigBee são feitas nas faixas de frequências que não precisam de licença (ISM). Tem-se então disponíveis as faixas de 2.4Ghz (mundialmente), 915Mhz (na América) e 868Mhz (na Europa), tendo associada uma taxa de transferência máxima de 250kbps na frequência de 2.4Ghz operando com 16 canais, 40kbps na frequência de 915Mhz operando com 10 canais e 20kbps na frequência de 868Mhz operando com 1 canal.

2.1.1 Tipos fundamentais de dispositivos ZigBee

- **Full Function Device (FFD)** - pode cumprir qualquer papel numa rede ZigBee, podendo desempenhar a função de coordenador de uma rede, *router* ou *end device*. Tratam-se de dispositivos de construção mais complexa, com boas capacidades de processamento e de memória;
- **Reduced Function Device (RFD)** - é limitado a uma configuração com topologia em estrela, não podendo actuar como um coordenador da rede. Pode comunicar com um coordenador ou com um *router*, não podendo trocar informação directamente com outro RFD. São dispositivos de construção mais simples e de baixo custo.

| Full Function Device | Reduced Function Device |
|---|--|
| <ul style="list-style-type: none"> • Ajuste de parâmetros da rede • Reencaminhamento de informações pela rede • Gestão dos nós da rede • Armazenamento de informações nos nós • Distribuição de mensagens entre nós da rede • Operação tipicamente no estado activo | <ul style="list-style-type: none"> • Função passiva na rede • Efectua buscas por redes disponíveis • Transferência de dados da aplicação • Determina o estado dos dados • Solicita dados ao coordenador da rede • Pode permanecer inactivo por longos períodos |

Tabela 2.2- Tabela de capacidades e funcionalidades dos dispositivos ZigBee.

Devemos observar que, em topologias com configuração estrela, uma rede ZigBee requer pelo menos um dispositivo FFD actuando como coordenador da rede, podendo os demais dispositivos serem do tipo RFD para reduzir o custo do sistema. A Figura 2.3 ilustra as possíveis topologias de uma rede ZigBee.

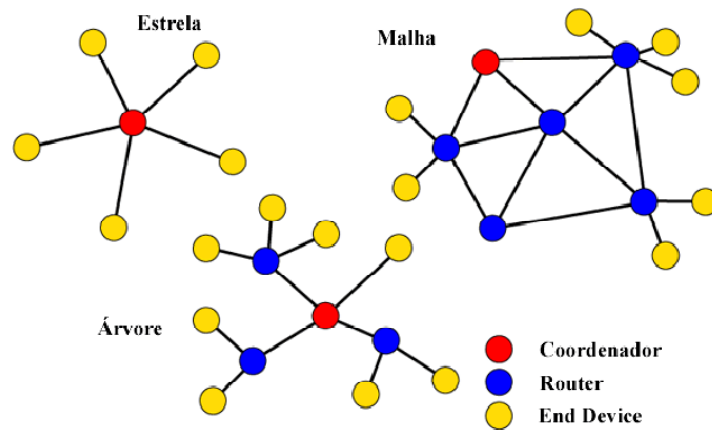


Figura 2.3 - Possíveis configurações de uma rede ZigBee.

2.1.2 Perfis de dispositivos ZigBee

Existem três diferentes tipos de perfis adoptados pelos equipamentos ZigBee, dependendo das suas capacidades e diferentes funções na rede:

- **ZigBee Coordinator (coordenador):** Há apenas um coordenador em cada rede, que é o dispositivo que tem o maior número de funções. O coordenador tem a capacidade de criar uma rede e é o único dispositivo capaz de comutar dados entre redes. Faz, também, o armazenamento das informações da rede;
- **ZigBee Router (routers):** São dispositivos que fornecem informações aos outros dispositivos da rede, fazendo o roteamento das comunicações como um *router* comum de redes IP;
- **ZigBee End Device (dispositivo final):** Tem apenas a função de trocar informações com um coordenador ou *routers*. Uma das suas vantagens é o facto de ser um dispositivo com pouca memória pois não precisa de rotear informações. São dispositivos de baixo custo.

2.1.3 Segurança na comunicação ZigBee

O padrão ZigBee adoptou a proposta de um novo algoritmo de segurança, baseado na simplificação do algoritmo de roteamento AODV (Ad-hoc On-demand Distance Vector). Esta proposta foi adoptada como parte da especificação IEEE 802.15.4. A camada MAC utiliza o padrão AES (Advanced Encryption Standard) como algoritmo de criptografia, descrevendo uma variedade de rotinas de segurança. Estas rotinas têm como objectivo garantir a confidencialidade, a integridade e a autenticidade das tramas da camada MAC. A camada MAC faz o processamento de segurança mas são as camadas superiores que controlam este processo, ajustando as chaves de

criptografia e determinando os níveis de segurança que deverão ser usados. Quando a camada MAC transmite/recebe uma trama, verifica o destino/fonte da trama, recupera a chave associada com esse destino/fonte e usa esta chave para processar a trama de acordo com a rotina de segurança designada. Cada chave é associada a uma única rotina de segurança e o cabeçalho da trama MAC possui um bit que indica se a segurança para a trama está habilitada.

2.2 Web services

Hoje em dia praticamente todos os computadores estão ligados a uma rede e as suas aplicações podem aproveitar as crescentes oportunidades para estabelecer relações dinamicamente. Para que tal seja possível as transacções têm de ser realizadas em tempo real e utilizar tecnologias padrão da Internet para integrar as diferentes ilhas de aplicações.

Os *web services* foram a solução encontrada para a integração de sistemas e na comunicação entre diferentes aplicações. Com esta tecnologia é possível que novas aplicações possam interagir com aquelas que já existem e que sistemas desenvolvidos em plataformas diferentes sejam compatíveis. Os *web services* são componentes que permitem às aplicações enviar e receber dados em formato XML, em que cada aplicação pode ter a sua própria "linguagem" interna, realizada de maneira compreensível, reutilizável e padronizada. É uma tentativa de organizar um cenário cercado por uma grande variedade de diferentes aplicativos, fornecedores e plataformas. No passado as empresas tinham receio de disponibilizar funcionalidades na Internet devido ao perigo de exporem os seus dados, mas com o desenvolvimento dos *web services* elas podem publicar serviços de forma simples e segura, estando totalmente isolados da base de dados. Para as empresas, os *web services* trazem agilidade para os processos e eficiência na comunicação entre cadeias de produção ou de logística. Toda e qualquer comunicação entre sistemas passa a ser dinâmica e, principalmente, segura. Acredita-se que no futuro as empresas irão listar os seus *web services* em directórios públicos (UDDI), de onde poderão ser vendidos como serviços para outras empresas, instituições ou utilizadores comuns.

2.2.1 Principais tecnologias utilizadas pelos *web services*

As bases para a construção de um *web service* são os padrões XML e SOAP. O transporte dos dados é realizado normalmente via HTTP (o padrão não determina o protocolo de transporte). Os dados são transferidos no formato XML e encapsulados pelo protocolo SOAP. As chamadas às operações, incluindo os parâmetros de entrada/saída, são codificadas no protocolo SOAP. Os serviços (operações, mensagens, parâmetros, etc.) são descritos usando WSDL - linguagem de descrição dos serviços disponibilizados. O processo de publicação, pesquisa e descoberta de *web*

services utiliza o protocolo UDDI, que é o repositório no qual podem ser publicados e localizados todos os *web services*.



Figura 2.4 - Interação entre um consumidor e um fornecedor de Web Services.

Assim, os serviços disponíveis podem ser descobertos pelos consumidores, possibilitando transações empresariais e comerciais pela rede, sendo este o princípio básico da arquitectura SOA, já mencionada anteriormente.

Em resumo, as principais características de um *web service* são:

- Baseado em XML;
- Fraco acoplamento entre o consumidor e o provedor do serviço;
- Capacidade de uma associação (ligação - *binding*) entre cliente e serviço de forma
- Síncrono ou assíncrono;
- Suporte à chamada de procedimento remoto (RPC - *Remote Procedure Call*);
- Suporte à troca de documentos.

2.3 .Net Framework

A .NET Framework da Microsoft é uma plataforma de desenvolvimento e execução de aplicações. Esta plataforma possui uma biblioteca com vários componentes pré-programados que facilitam a criação e desenvolvimento de software que corre em máquinas que possuam a plataforma instalada.

2.3.1 Windows Workflow Foundation

O Windows Workflow Foundation (WF) é uma tecnologia que faz parte da .NET Framework 3.0/3.5 e que tem como função definir, gerir e controlar as várias fases de desenvolvimento de um projecto [19]. Esta tecnologia permite a separação entre a lógica das aplicações e os vários componentes de execução subjacentes, possibilitando assim uma visão da aplicação que simplifica a sua gestão. A maior parte dos processos de negócio possuem um modelo que os define, quer seja um fluxograma, um diagrama UML ou casos de utilização. Nestes processos existe um fluxo de actividades que, ocorrendo de forma sequencial ou concorrente, trocam informação de forma a cumprir um objectivo comum. A coordenação destas tarefas, efectuadas por humanos ou máquinas, tem de ser gerida de forma óptima. O planeamento de um projecto envolve normalmente a elaboração de diagramas, fluxogramas e pseudo-código. O WF possibilita a criação de um modelo detalhado da integração de várias aplicações e da forma como elas interagem e contribuem para o projecto do qual fazem parte. A grande vantagem do WF é o facto de que ao construir este modelo está-se também a construir a aplicação.

2.3.2 Windows Presentation Foundation

O Windows Presentation Foundation (WPF) é o subsistema gráfico do .NET Framework 3.0/3.5, que possibilita um modelo de programação para construção de aplicações no qual existe uma total separação entre o desenvolvimento da interface e da lógica de programação [20]. Uma aplicação WPF pode ser distribuída autonomamente ou via *browser*, sendo que neste último modo as aplicações desenvolvidas, *XAML Browser Applications*, estão sujeitas a restrições, como indicado na Tabela 2.3. O WPF assenta fundamentalmente no XAML, uma linguagem declarativa baseada no XML, utilizada para definir elementos da interface como objectos 2D/3D e todo o tipo de animações e eventos associados. Tudo o que é criado ou implementado em XAML pode ser expresso em linguagens .NET mais tradicionais, como o C# e o Visual Basic .NET, embora a programação em XAML permita um nível de abstracção e de simplicidade superiores. Um elemento chave desta linguagem é a reduzida complexidade das ferramentas que processam o código criado por esta tecnologia.

Em resumo, as vantagens do WPF são:

- Simplicidade na construção de interfaces;
- Animações avançadas;
- Fácil integração de imagens, áudio e vídeo;
- Implementa DirectX;
- Aproveita as capacidades das novas placas gráficas.

| WPF | XBAP |
|--|--|
| <ul style="list-style-type: none"> • Instalada no computador do utilizador • Corre com todas as permissões • Acesso total ao disco • Implementa completamente o WCF • Utilizador pode estar <i>online</i> ou <i>offline</i> • A versão utilizada pode não ser a mais recente | <ul style="list-style-type: none"> • Não instalada no computador do utilizador • Corre com a segurança definida no <i>browser</i> • Acesso isolado ao disco • Implementa parcialmente o WCF • Utilizador necessita de estar <i>online</i> • A versão utilizada é sempre a mais recente |

Tabela 2.3 - Características das aplicações WPF e XBAP.

2.3.3 Windows Communication Foundation

O Windows Communication Foundation (WCF) é o subsistema do .NET Framework 3.0/3.5 dedicado à criação de aplicações que comunicam entre si [21]. As aplicações WCF podem utilizar mensagens SOAP na comunicação entre processos, tornando-as compatíveis com todos os outros processos que suportem esta linguagem.

Quando um processo WCF comunica com um processo não-WCF é utilizada codificação baseada em XML nas mensagens SOAP transmitidas. Por outro lado, quando dois processos WCF comunicam entre si as mensagens SOAP são codificadas num formato binário optimizado, estando ambas as codificações em conformidade com a estrutura de dados do formato SOAP, denominada *Infoset*. O WCF foi desenvolvido em conformidade com os princípios de uma arquitectura orientada a serviços, pelo que suporta processamento distribuído onde os serviços são utilizados por vários clientes. Um cliente pode consumir vários serviços, e estes podem ser consumidos por vários clientes. Os serviços possuem, geralmente, um interface WSDL que qualquer cliente WCF pode utilizar para consumir o serviço, indiferenciadamente da plataforma que o disponibiliza. O WCF implementa os principais padrões WS (web services), como o WS-Addressing, o WS-ReliableMessaging e o WS-Security, que asseguram a fiabilidade e segurança das comunicações.

2.4 Redes neuronais

Uma das áreas mais fascinantes da investigação é aquela que se ocupa com o estudo do cérebro. Foi o interesse nesta área que ditou os primeiros avanços nos campos da inteligência artificial, redes neuronais e aprendizagem de máquina. O estudo do cérebro humano levou à descoberta de que a informação em sistemas neuronais biológicos é processada de forma paralela numa rede com um elevado número de neurónios interligados, que actuam como pequenas unidades computacionais. Há, contudo, várias diferenças entre o modo de funcionamento do cérebro humano e uma rede neuronal artificial (RNA). Uma dessas diferenças é o facto da saída de um nó de uma RNA ser um valor único e constante, enquanto um neurónio biológico transmite uma série temporal, geralmente complexa, de impulsos [2][3]. As redes neuronais artificiais são geralmente aplicadas à regressão e classificação de dados, após um período de aprendizagem. As suas principais características são:

- São constituídas por pequenas e idênticas unidades computacionais - os nós, que actuam como neurónios artificiais;
- As ligações entre nós têm um peso associado;
- Os pesos de uma RNA são ajustados durante o processo de aprendizagem;
- São geralmente utilizadas transformações não lineares dos valores associados a cada nó.

Os neurónios, e os seus equivalentes artificiais, os nós, recebem estímulos de neurónios à sua volta. Esses estímulos podem ser positivos (excitação) ou negativos (inibição), e quando a combinação de todos esses estímulos ultrapassa um determinado limiar, esse neurónio entra em acção. Numa RNA estes estímulos são vectores n -dimensionais de números reais, em que cada uma das componentes deste vector tem um peso associado, que é um número real positivo ou negativo. As Redes Neuronais Artificiais (RNAs) funcionam de forma análoga ao cérebro humano, com um processo de aprendizagem igualmente constituído por erros e descobertas, adquirindo conhecimento através da experiência. São constituídas por camadas e por nós que estão ligados entre si, formando uma rede que graças ao processamento paralelo da informação tem um elevado desempenho. Estas redes são sistemas de processamento de informação que empregam algumas características dos processadores neuronais biológicos. São capazes de analisar dados através da passagem de informação por várias unidades neuronais que estão interligadas numa estrutura altamente distribuída, que exhibe um comportamento global complexo.

Uma RNA aceita entradas x_i , que são depois multiplicadas por um conjunto de pesos w_i . Um nó transforma a soma ponderada num valor de saída y , de forma não-linear, usando uma função de activação, ou transferência, T .

Se a soma pesada dos estímulos:

$$\sum_{i=1}^n w_i x_i = x^T w \quad (2.1)$$

for superior a um dado limiar então o nó fica activo.

As funções de activação mais utilizadas são:

| | |
|----------------------|---------------------------------|
| Limite | $Se\ x > 0, g(x) = 1$ |
| Linear | $g(x) = x$ |
| Sigmoidal | $g(x) = \frac{1}{(1 + e^{-x})}$ |
| Tangente hiperbólica | $g(x) = \tanh x$ |

Tabela 2.4 - Funções de activação mais utilizadas em redes neuronais.

As redes são constituídas tipicamente por uma camada de entrada, uma camada escondida e uma camada de saída. A camada de entrada recebe os estímulos e passa-os, de acordo com o tipo de ligação, à camada escondida que por sua vez os transmite à camada de saída, usando a função de activação escolhida. À medida que a rede aprende uma tarefa, as ligações sinápticas entre os nós é reforçada. Cada ligação entre os nós possui um peso associado, que é actualizado após avaliação da diferença entre o resultado obtido e o resultado esperado.

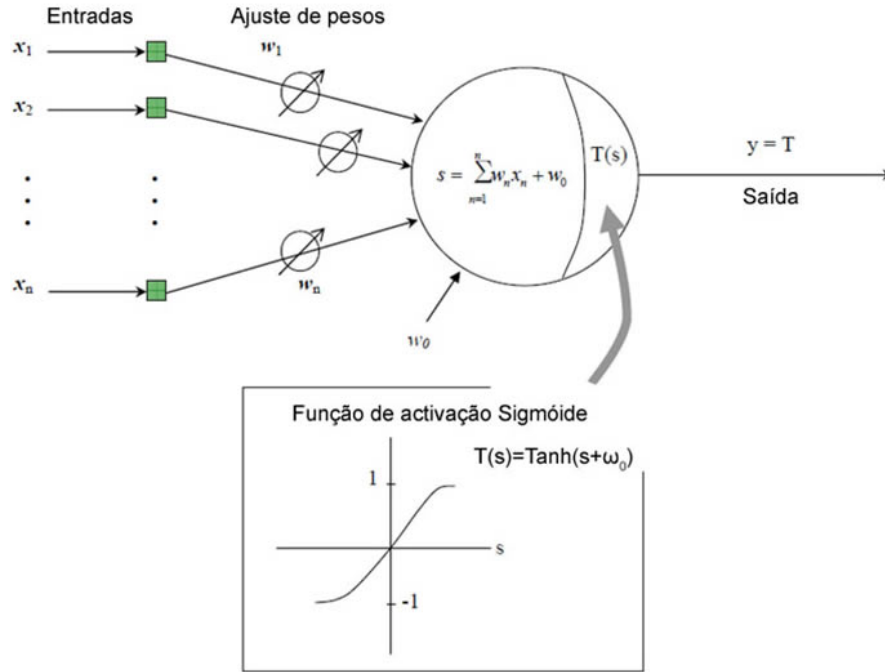


Figura 2.5 - Funcionamento de uma rede neuronal.

Os nós da camada escondida são análogos ao interneurónios do cérebro, onde as correlações entre a informação de entrada e a informações de saída são capturadas. Isto permite à rede aprender, ajustar e generalizar, tendo como base a aprendizagem prévia. Nas camadas escondidas, cada nó processa a soma ponderada das suas entradas para formar um sinal de activação, denominado aqui por *net*. Assim, este sinal é o produto interno das entradas com os pesos do nó escondido.

Considerando uma entrada $x_0 = 1$, por simplicidade, e um peso associado ao nó igual a w_0 , pode-se dizer que para o j – ésimo neurónio:

$$net_j = \sum_{i=1}^d x_i w_{ij} + w_{j0} = \sum_{i=0}^d x_i w_{ij} \equiv w_j^T x \quad (2.2)$$

onde w_{ij} representa o peso da ligação do i – ésimo nó da camada anterior ao j – ésimo nó da camada actual. A saída deste j – ésimo nó é uma função não-linear - sigmoideal - do seu sinal de activação net_j , e é igual a :

$$y_j = f(net_j) = f\left(\sum w_j^T x\right) \quad (2.3)$$

É assumido que essa mesma não-linearidade é utilizada em todos os nós escondidos. Cada nó de saída processa de forma idêntica o seu sinal de activação, baseado nas saídas dos nós da camada escondida, e obtêm-se os seguintes valores de saída:

$$z_k = f(net_k) = f\left(\sum w_j^T x\right) \quad (2.4)$$

Esta não-linearidade na saída pode ser, por exemplo, uma função seno.

2.4.1 Arquitectura das redes

Determinar a arquitectura ideal de uma RNA para modelar um problema - maximizando o binómio *velocidade da rede/eficiência dos resultados* - não é um processo simples, pois não existe um procedimento definido que torne possível saber à partida qual o número de camadas ou nós escondidos que permitem otimizar a solução, o mesmo se passando relativamente aos vários parâmetros da rede. A metodologia a adoptar passa por estimar um ponto de partida e otimizar progressivamente a resposta da rede pelo método tentativa/erro.

Foi demonstrado que reduzir ao máximo o número de nós da rede, mantendo o erro dentro dos limites pretendidos, é desejável [6][16], devido às seguintes razões:

- Reduz significativamente o processamento computacional e aumenta consequentemente a velocidade de resposta da rede;
- Ajuda a evitar o excesso de treino, que limita a capacidade de generalização da rede, pois são aprendidas as particularidades das amostras de treino;
- É mais fácil analisar a evolução da rede, e detectar possíveis problemas, com um número reduzido de nós.

Para uma rede com uma única camada escondida a seguinte abordagem permite definir um bom ponto de partida para a arquitectura da rede [3][16]:

Se n_e representa o número de nós escondidos e n o número de nós de entrada, então:

- n_e deve ser aproximadamente 75% do número de nós da camada de entrada;
- n_e deve estar entre a média e a soma do número de nós das camadas de entrada e saída;
- Devem-se inicialmente utilizar $2n + 1$ nós escondidos e iterativamente reduzir o número de nós escondidos até o tempo de resposta corresponder ao pretendido. Em alternativa pode utilizar-se um número reduzido de nós de entrada e progressivamente introduzir mais nós até a resposta possuir um erro inferior ao limite aceite;

- Uma camada escondida com $n_e = n + 1$ garante que não se incorre num mínimo local, situação em que a rede não consegue evoluir no sentido de otimizar os resultados.

2.4.2 Capacidade de representação

O poder de representação de uma rede neuronal aumenta de acordo com o crescimento do número de camadas escondidas, pois estas determinam a capacidade de modelar funções não-lineares complexas.

A expressividade de uma rede, baseada no número de camadas escondidas, pode ser sumariada da seguinte forma [2]:

- Uma camada escondida: qualquer função contínua pode ser representada, desde que o número de nós escondidos seja suficiente;
- Duas camadas escondidas: qualquer função contínua ou convexa aberta pode ser representada. O principal inconveniente de utilizar duas ou mais camadas escondidas é o aumento do processamento computacional requerido, traduzindo-se numa redução da velocidade da rede que poderá ser incompatível com a exigência da aplicação associada. A utilização de duas ou mais camadas aumenta igualmente a probabilidade de se ficar preso num mínimo local;
- Três camadas escondidas: Qualquer função arbitrária pode ser representada, sendo a sua complexidade limitada pelo número de nós utilizados.

2.4.3 Treino das redes neuronais

As RNAs desenvolvem uma estrutura interna que lhes permite resolver problemas ou efectuar previsões. Através do processo de treino os pesos das ligações rearranjam os seus valores e revelam um padrão de dados. O que distingue uma rede neuronal de outras ferramentas de classificação clássicas é o facto de uma RNA ser treinada e não programada.

À medida que cada conjunto de entradas/saídas é apresentado à rede, o mapeamento interno é registado na camada escondida, o que confere à rede inteligência e uma capacidade de previsão intuitiva. A aprendizagem da rede é conseguida com cada par entrada/saída apresentado, ajustando o seu comportamento a cada informação aprendida. A saída é calculada de acordo com os valores de entrada e os pesos das ligações da rede, que podem ser interpretados como a função de transferência da rede.

É boa prática relacionar o número de amostras disponíveis para treinar a rede com o número de nós escondidos, sendo o número de amostras sempre inferior ao número de nós da camada escondida. Desta forma evita-se o treino excessivo e assegura-se uma boa capacidade de generalização. O rácio sugerido para a relação *número de nós escondidos/número de amostras de*

treino é de 2, 4, 10 e 30, segundo diferentes autores. Alguns métodos que permitem determinar uma arquitectura próxima da ideal são o adaptativo, o de correlação em cascata e os algoritmos genéticos.

Há dois tipos de aprendizagem:

- Não-supervisionada: Apenas as entradas são apresentadas à rede. Não há valores de saída concretos que permitam o ajuste dos pesos das ligações. Este método de treino tenta identificar padrões;
- Supervisionada: São apresentados à rede conjuntos de valores de entrada/saída. Consequentemente os pesos das ligações, inicialmente distribuídos aleatoriamente, ajustam os seus valores para produzir valores que sejam o mais aproximados possível dos valores de saída apresentados. A cada iteração de aprendizagem espera-se que o erro entre a saída desejada e a obtida diminua. O resultado que é possível obter é um erro minimizado entre estas, bem como da estrutura interna da rede que representa a dependência entrada/saída.

2.4.4 Rede de retropropagação

Uma rede de retropropagação é uma RNA que emprega o algoritmo de retropropagação, o tipo mais utilizado de aprendizagem supervisionada, para solucionar problemas de reconhecimento de padrões complexos e efectuar funções de mapeamentos não triviais. Esta rede foi formalizada inicialmente por Werbos e mais tarde por Parker e por Rumelhart e McClelland [2].

Inicialmente, a rede aprende um conjunto de pares de exemplo entrada/saída utilizando um ciclo propagação-adaptação de duas fases. Após um padrão de entrada ser aplicado como estímulo à camada de entrada da rede, os sinais gerados são propagados até à camada de saída. Este padrão de saída é então comparado com o vector de saída desejado e é gerado um valor de erro para cada unidade de saída. Estes valores de erro são transmitidos aos nós da camada escondida anterior, directamente ligada a esta, advindo daí o nome do algoritmo: retropropagação. No entanto, cada unidade desta camada escondida recebe apenas uma porção do valor do erro total, dependendo da contribuição de cada unidade para o valor de saída. Este processo é repetido, camada a camada, até cada unidade da rede ter recebido um valor que traduz a sua contribuição relativa para o erro total. Baseando-se neste valor de erro recebido, cada unidade actualiza os pesos das suas ligações, de forma a que a rede possa convergir para um estado que permita que todos os padrões de treino possam ser codificados. A importância deste processo é o facto de, com o decorrer do treino, os nós das camadas intermédias organizam-se de forma a que diferentes nós aprendam a reconhecer diferentes características de todo o espaço

dos valores de entrada. Após o treino, quando estimuladas com um padrão de entrada arbitrário que possua ruído ou está incompleto, as unidades das camadas escondidas respondem com um valor de saída activo se os valores de entrada possuírem características similares a um dos padrões de treino.

Durante o treino as RRP tendem a desenvolver relações internas entre os nós de forma a organizar as amostras de treino em classes de padrões. Estas relações são o que possibilita que, quando confrontadas com as amostras utilizadas como treino, a rede apresente como resposta a saída desejada. Esta mesma representação interna pode ser aplicada a entradas nunca antes aprendidas durante o processo de treino, sendo que nesta circunstância a rede irá classificar estas novas entradas segundo as semelhanças que elas partilham com as amostras aprendidas.

2.4.5 Regra delta generalizada

Uma rede de retropropagação utiliza como algoritmo de aprendizagem a regra Delta generalizada. Esta regra expressa um procedimento recursivo que permite atribuir a cada ligação a responsabilidade pelo erro causado nas unidades de saída da rede [5][10]. Os pesos daquelas unidades podem assim ser ajustados de forma a que esse erro seja minimizado. Se existir uma diferença entre a saída desejada e a obtida então os pesos necessitam de ser ajustados. Matematicamente, os elementos que processam a informação da rede precisam de ser modificados de forma a poderem monitorar o seu valor de saída. Em seguida, o valor de saída actual é comparado com a saída do modelo z_k . O erro e_k é processado para o padrão de entrada x_k como a diferença entre a saída actual y_k e z_k , $e_k = y_k - z_k$.

O método de aprendizagem de retropropagação começa por enviar o padrão de treino para a camada de entrada, passa os sinais pela rede e determina os valores de saída. Os k valores de saída z_k são depois comparados com os valores de saída esperados y_k e a sua diferença é tratada como um erro, como acima descrito. A seguinte função de critério:

$$E(w) = \frac{1}{2} \sum_{i=1}^k e_i^2 = \frac{1}{2} \|e\|^2 \quad (2.5)$$

onde:

$$e = [e_1, e_2, \dots, e_k]^T \quad (2.6)$$

ou, para uma série de Q amostras de treino numa época:

$$E(w) = \frac{1}{2} \sum_{q=1}^Q \|e^q\|^2 \quad (2.7)$$

é minimizada no que diz respeito ao peso das ligações, onde y e z representam respectivamente a saída esperada e a saída actual.

$$E(w) = \frac{1}{2} \|e\|^2 = \frac{1}{2} \sum_{i=1}^k (y_i - z_i)^2 = \frac{1}{2} \|y - z\|^2 \quad (2.8)$$

A regra da retropropagação é baseada no algoritmo da descida do gradiente. O vector peso w em cada camada é inicializado com valores aleatórios que são iterativamente actualizados no sentido de reduzir o erro E :

$$w(i+1) = w(i) + \Delta w(i) \quad (2.9)$$

$$\Delta w = -\eta \frac{\partial E}{\partial w} \quad (2.10)$$

ou, explicitando as componentes:

$$\Delta w_{pq} = -\eta \frac{\partial J}{\partial w_{pq}} \quad (2.11)$$

onde η é o passo de aprendizagem ($0 < \eta < 1$). A iteração k representa a k – ésima amostra do padrão...(ou o k – ésimo padrão numa série de treino).

Considere o vector peso na l – ésima camada, para o peso w_{ij}^l na ligação entre o seu j – ésimo nó e o i – ésimo nó da camada $(l-1)$ – ésima:

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial net_j} \frac{\partial net_j}{\partial w_{ij}^l} = \delta_j^l y_i^{l-1} \quad (2.12)$$

onde y_i^{l-1} representa a saída do i – ésimo neurónio da $(l-1)$ – ésima camada. Para o nó de saída j :

$$\delta_j^q = \frac{\partial E}{\partial net_j} = -(t_j - z_j) f'(net_j) \quad (2.13)$$

e para o nó escondido j na camada l :

$$\delta_j^q = \frac{\partial E}{\partial net_j} = f'(net_j) \sum_k \delta_k^{l+1} w_{jk} \quad (2.14)$$

Pelo menos três camadas são necessárias: entrada, escondida e saída. Embora o número de nós escondidos deva ser grande, não se deve privar a rede da sua capacidade de generalização quando a rede começa a memorizar em vez de deduzir. Por outro lado, se a camada escondida for pequena a capacidade de memória é restringida.

O algoritmo de retropropagação segue os seguintes passos [17]:

1. Inicialização: inicializa os pesos das ligações e os neurónios com valores aleatoriamente baixos;
2. Introdução de dados: introduz o conjunto de dados entrada/saída que serão utilizados pela rede para a aprendizagem;
3. Cálculo da saída: calcula os valores de saída e ajusta os pesos das ligações aplicando o erro actual da rede;

4. Ajustes: Ajusta os limites de activação e os pesos dos neurónios da camada escondida de acordo com a regra Delta;
5. Repete os passos 2 a 4 para convergir para o erro mínimo.

Optimizar uma rede neuronal é um processo demorado, pois existem vários parâmetros, que serão em seguida apresentados e explicados.

2.4.6 Passo de aprendizagem

A regra de actualização dos pesos de uma ligação é:

$$W_{ji} \leftarrow W_{ji} + \eta \times \delta_i \times o_j \quad (2.15)$$

onde η é o passo de aprendizagem e desempenha um papel muito importante. Representa a velocidade de alteração do peso de cada ligação. As redes neuronais fazem a pesquisa por um espaço de peso na procura da melhor configuração. Este valor representa a unidade de variação nesta pesquisa. Se o passo for pequeno a convergência será lenta, pois serão precisas muitas iterações para se atingir a melhor configuração. No caso de o passo ser um valor elevado existe a possibilidade de a solução ficar retida num mínimo local.

Não é garantida a convergência de uma rede multicamada para a solução óptima, pelo que escolha adequada deste parâmetro é da maior importância.

2.4.7 Critério de paragem

O critério de paragem determina quando o treino de uma rede neuronal deve ser concluído. Alguns métodos que definem esta paragem são [3]:

- Treinar a rede durante um determinado número de épocas;
- Treinar até que o erro se torna inferior ao valor pretendido ou deixa de apresentar variações significativas;
- Treinar até à variação de peso das ligações se tornar imperceptível.

Uma abordagem aconselhada é dividir o conjunto de amostras utilizadas para teste em dois conjuntos: um subconjunto de treino e um subconjunto de validação. O erro em resposta ao subconjunto de validação pode ser utilizado como critério de paragem em vez do erro no subconjunto de treino, como forma de prevenir o treino excessivo, como se exemplifica na Figura 2.6. É, contudo, necessária precaução na decisão de paragem de treino pois não se garante que o erro em resposta ao subconjunto de validação decresça monotonicamente.

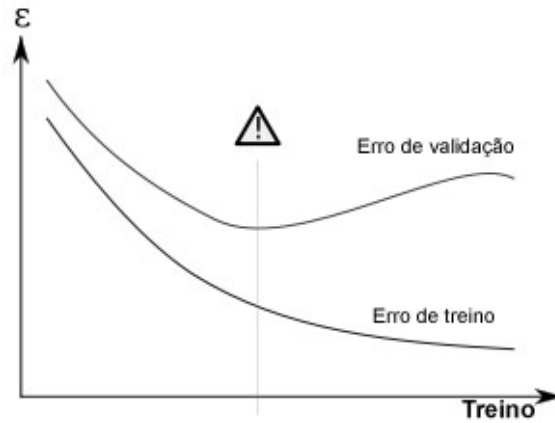


Figura 2.6 - Exemplo de sobretreino.

2.4.8 Momento

Como mencionado anteriormente, uma rede neuronal pesquisa no espaço dos pesos em busca da combinação óptima de pesos das ligações. Um dos problemas mais comuns nessa pesquisa é atingir um ótimo local. Uma das técnicas comuns é adicionar um factor, que representa o momento, à fórmula de actualização de peso, que permite ultrapassar mínimos locais.

A equação de actualização dos pesos é modificada da seguinte forma, para contemplar o factor de momento [3]:

$$W_{ji} \leftarrow W_{ji} + \eta \times \delta_i \times o_j + \alpha \Delta W_{ji(\text{anterior})} \quad (2.16)$$

onde α representa o momento. Para além de evitar mínimos locais, o momento ajuda também a aumentar a velocidade de convergência da rede para uma solução óptima.

O correcto ajuste do factor de momento é importante e é conseguido, mais uma vez, através do método tentativa/erro. Um valor de α muito elevado pode impedir a rede de encontrar o ótimo global ou mesmo um bom ótimo local.

Capítulo 3

Trabalho Relacionado

Os ambientes inteligentes têm sido idealizados há várias décadas, mas só recentemente têm-se tornado realidade. O maior impedimento à sua integração nas nossas habitações é a falta de aceitação por parte dos utilizadores que não consideram necessárias as vantagens que estes sistemas disponibilizam, além de considerá-los economicamente inviáveis e tecnicamente falíveis. Os projectos mais relevantes e que obtiveram melhores resultados foram os projectos MavHome e Amigo, que são apresentados em seguida.

3.1 MavHome

O projecto MavHome teve como foco a investigação sobre tecnologias de domótica que permitissem tratar o ambiente habitacional como um agente inteligente [11][12]. Teve como principais objectivos maximizar o conforto dos habitantes, minimizar o consumo de recursos e manter a segurança da habitação. A tecnologia X10 foi a escolhida para comunicação entre equipamentos. Como ambiente de teste foi utilizada uma habitação, durante vários meses, onde várias pessoas simulavam as actividades do dia-a-dia, que eram aprendidas usando Modelos Escondidos de Markov, após tratamento dos dados.

O ambiente Mavhome é composto por agentes autónomos que estão estruturados hierarquicamente. Um agente pode estar responsável por um frigorífico enquanto outro activo o sistema de rega. Esta hierarquia dita quais agentes têm maior poder sobre as decisões.

A arquitectura de cada agente é basicamente composta por quatro camadas:

- Camada de Decisão: Recebe informação que depois de tratada é aprendida. Usa este conhecimento na tomada de decisão de acções, autónomas, que visem o interesse da habitação e dos seus ocupantes;
- Camada de Informação: Agrega, guarda e gera conhecimento necessário para a tomada de decisões. Contém e disponibiliza componentes de previsão, bases de dados, interfaces, sintetizadores de voz e componentes de reconhecimento;
- Camada de Comunicação: Disponibiliza a todas as camadas um meio de comunicação fácil de usar, e permite a descoberta de serviços entre componentes de diferentes camadas;
- Camada Física: Gere a comunicação entre todos os equipamentos, como sensores, actuadores, computadores e equipamentos de rede.

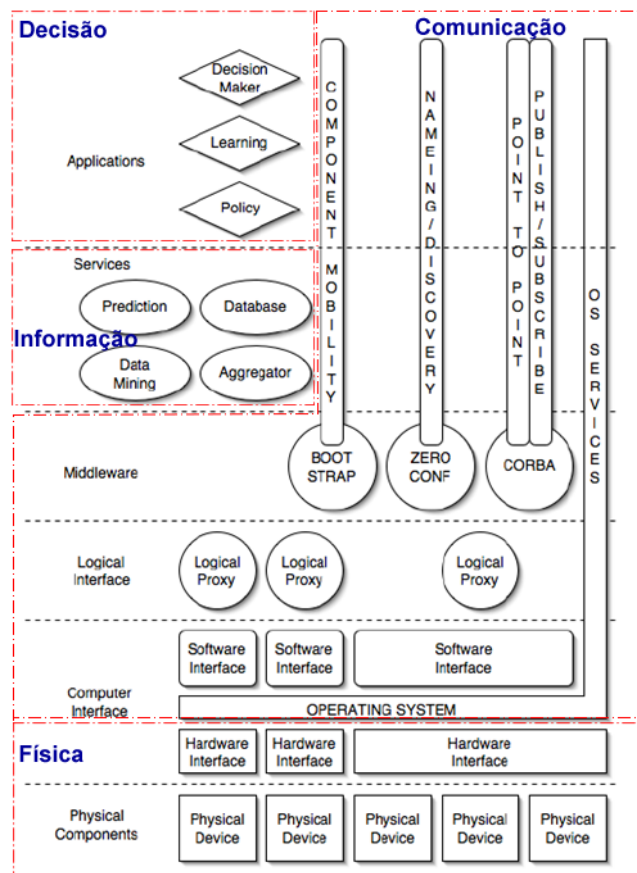


Figura 3.1 - Arquitectura completa do MavHome.

A percepção tem lugar na camada inferior, com a camada de decisão a notificar a camada de comunicação de alterações no valor de sensores, que por sua vez alerta outros agentes que estejam interessados nesta informação. A camada de informação guarda o estado e a camada de decisão escolhe a acção apropriada com base na informação.

3.1.1 Algoritmos de previsão

Neste projecto foram utilizados conjuntamente três algoritmos para a previsão das acções a tomar. Eles são:

- Previsão por padrões de sequências: Este algoritmo, denominado SHIP, foi desenvolvido pela equipa que implementou este projecto. O seu funcionamento consiste em analisar a mais recente sequência de eventos e encontrar semelhanças com sequências prévias. Quando um utilizador executa um comando num dispositivo, esta acção é registada num historial de acções. A detecção de padrões é feita concorrentemente, de forma a garantir um funcionamento em tempo-real;
- Previsão baseada na compressão: Denominado Active LeZi (ALZ), este algoritmo usa princípios da teoria da informação no processamento de sequências de actividades passadas. Devido à interacção habitante-dispositivo ser caracterizada como uma cadeia de eventos Markov é possível utilizar um esquema de previsão de sequências que é óptimo em termos de precisão na previsão para este tipo de problemas;
- Previsão por um modelo de Markov baseado em tarefas: Este algoritmo identifica tarefas de alto nível em sequências de acções para auxiliar a criação de um modelo de Markov. Um modelo de Markov simples pode ser gerado a partir de uma colecção de sequências de acções, sendo depois utilizado na previsão de acções futuras, sabendo o estado actual.

3.1.2 Experiências e testes

Foram conduzidos testes para determinar se o sistema era capaz de automatizar algumas das tarefas dos habitantes. Foram recolhidos dados sobre três actividades (ver TV, ouvir música através de um leitor de CD, e leitura de livros), que foram posteriormente extrapolados para o período de trinta dias, em que estas actividades ocorriam em diferentes alturas do dia, definidas aleatoriamente. Os testes foram realizados por um utilizador, que efectuou cada uma destas actividades quatro vezes, num período de duas horas. O sistema conseguiu automatizar dez das doze actividades efectuadas, o que se traduz numa taxa de sucesso de 83,3%. As falhas foram

originadas por ruído nos dados recolhidos por sensores, que quando processados pela sistema de aprendizagem não foram reconhecidos.

Actualmente estão a ser conduzidos testes em quatro novos cenários, com vista ao aperfeiçoamento e futura integração das soluções encontradas.

3.2 Amigo

O projecto Amigo é um projecto co-fundado pela União Europeia que teve início em Setembro de 2004 e fim em Março de 2008. Este projecto contou com a participação de 15 parceiros, entre os quais a Philips, a Fraunhofer, o INRIA e a Telefónica.

O projecto Amigo consistiu no desenvolvimento de *middleware* que dinamicamente integra sistemas heterógeneos para atingir a interoperabilidade entre dispositivos e serviços. Esta interoperabilidade pode também ser estendida a outras habitações.

A sua arquitectura orientada a serviços consiste em [1]:

- Camada *Middleware* de Base: Possui toda a funcionalidade necessária para suportar uma rede ambiente habitacional. Ela disponibiliza a semântica para descobrir e comunicar com serviços disponíveis na rede, incluindo aqueles que são baseados em padrões de comunicação e descoberta como o UPnP, o WS ou o SLP. São disponibilizados mecanismos de segurança para autenticação, autorização e encriptação;
- Camada de Serviços Inteligentes de Utilizador: Possui a funcionalidade necessária facilitar a interligação entre utilizadores e provedores de serviços. Providencia informação contextual, combina múltiplas fontes de informação e efectua previsões baseadas em padrões. A informação é adaptada aos perfis dos utilizadores a responde à situação e variações dos utilizadores no contexto;
- Plataforma de programação e de distribuição: Contém módulos que facilitam o desenvolvimento de serviços, .NET ou Java, que permitem a interligação de equipamentos. Disponibilizam suporte para interoperabilidade, segurança e descrição de serviços. Permite a abstracção dos protocolos usados para a descoberta e comunicação, facilitando a implementação. Dessa forma, serviços heterógeneos podem ser integrados na rede habitacional de forma independente das tecnologias de hardware e software subjacentes.

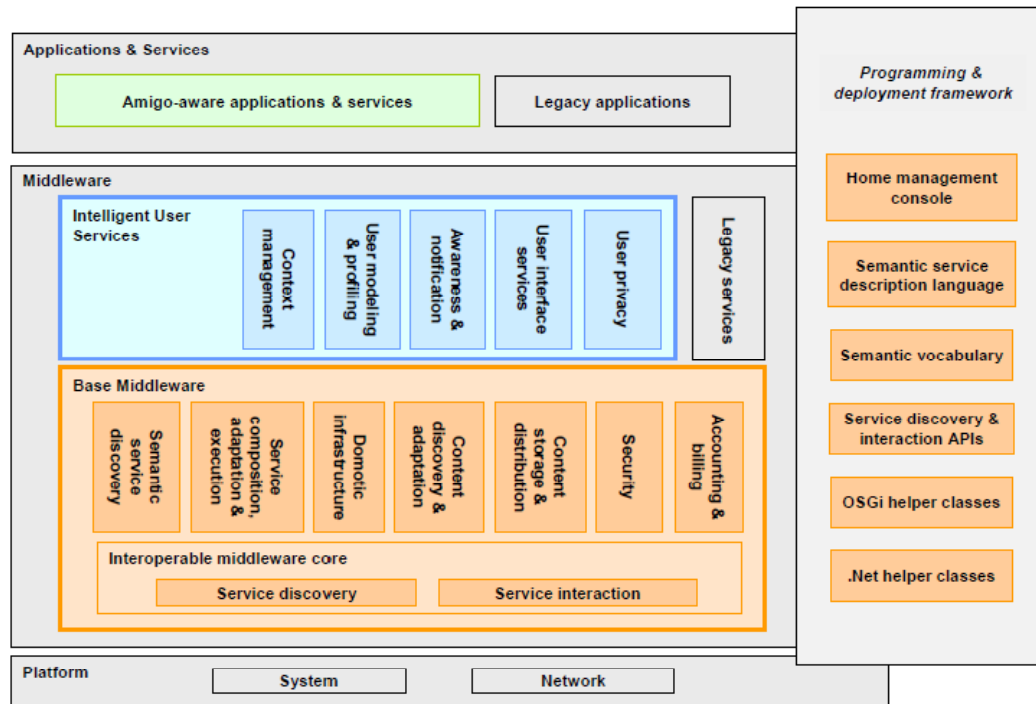


Figura 3.2 - Arquitectura completa do projecto Amigo.

Entre os serviços inteligentes de utilizador disponibilizados os de maior relevo são:

- Serviço de Gestão de Contexto: é uma infraestrutura aberta para gestão da informação contextual. O seu papel é o de adquirir informação vinda de várias fontes tais como sensores, actividades de utilizadores, aplicações em processos ou aplicações de Internet e subsequentemente combinar ou abstrair estas informações em “informação contextual”, para ser providenciado a serviços que solicitem esta informação;
- Monitorização e Notificação: disponibiliza a funcionalidade básica necessária para desenvolver aplicações que permitam a outras aplicações, ou pessoas, serem informadas sobre qualquer alteração significativa no contexto. As aplicações registam regras de monitorização que especificam de que tipo de mudanças devem ser notificadas, e com que nível de importância, com base nas preferências dos utilizadores ou condições pré-registadas;
- Modelação e Catalogação: disponibiliza a metodologia para optimizar a eficiência e usabilidade de serviços e interfaces de forma a:
 - a. Adaptar a apresentação da informação às preferências dos utilizadores;
 - b. Inferir sobre o comportamento futuro dos utilizadores;
 - c. Ajudar os utilizadores a encontrar informação relevante;

- d. Adaptar as características dos interfaces aos utilizadores e aos contextos onde estão inseridos;
- e. Indicar características dos interfaces e de apresentação da informação para a adaptação a ambientes multi-utilizador.

No projecto Amigo a aprendizagem de máquina é conseguida através do uso de Máquinas de Vectores de Suporte. Sendo um projecto recente, ainda não foi disponibilizada informação sobre a qualidade dos resultados obtidos.

3.3 Resumo

Nenhum dos projectos apresentados utilizou redes neuronais para a aprendizagem e previsão das actividades dos ocupantes da habitação. No caso do projecto Amigo foram usadas as máquinas de vectores de suporte (MSV), um método igualmente determinístico. Uma das vantagens das MSV é o facto de a sua solução ser sempre global e única, ao contrário das RNA que possuem mínimos locais. Outra das vantagens das MSV é a sua simples interpretação geométrica.

No projecto MavHome foram utilizados algoritmos orientados à detecção de sequências, como o ALZ e cadeias de Markov, com bons resultados. Estes algoritmos identificam padrões de acções consecutivas e associam-nos a um historial de actividades passadas, permitindo assim identificar a rotina actual.

Capítulo 4

Arquitectura do Sistema

Nesta capítulo serão indicados os requisitos necessários para que a solução implementada cumpra os objectivos pretendidos. São também explicadas as várias decisões tomadas ao longo do desenvolvimento do projecto.

4.1 Arquitectura

No desenvolvimento do módulo de inteligência, denominado Rede Neuronal HomeSim, teve-se a preocupação de assegurar uma integração fácil no sistema de controlo anteriormente criado. O simulador, a aplicação HomeSim, foi implementado de forma a também funcionar de forma autónoma, sem a comunicação com o módulo de inteligência, embora assim não seja possível usufruir de todo o seu potencial. Para a análise de todos os parâmetros da rede foi criado a Rede Neuronal Visual, uma aplicação que facilita a interpretação do funcionamento da rede e a sua optimização.

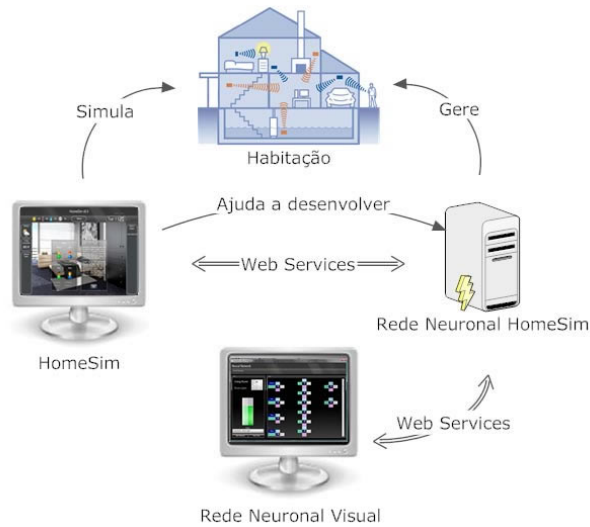


Figura 4.1 - Relação entre as diferentes aplicações.

4.2 Requisitos do Sistema

Para que o simulador possa ser facilmente acedido ele deve estar disponível *online*, de forma a que vários utilizadores possam testar a aplicação e assim contribuir para o seu desenvolvimento. O acesso *online* aplica restrições ao tipo de aplicação distribuída, com principal foco na compatibilidade e tamanho. Como o HomeSim corre na máquina de cada utilizador, a aplicação tem de comunicar com o módulo de inteligência de uma forma segura e fiável. Para que haja apenas uma habitação simulada existe apenas uma instância do Rede Neuronal HomeSim, que regista todas as actividades e monitoriza as condições da habitação. Desta forma o cenário visualizado é comum a todos os utilizadores, reproduzindo o conceito de espaço partilhado.

Para que a solução consiga reproduzir as preferências do utilizador, nomeadamente em termos de intensidade luminosa e temperatura ambientes dos espaços, é fundamental contemplar as condições ambientais, pela influência directa ou indirecta que exercem sobre as acções que tomamos, principalmente ao nível da iluminação e temperatura. Por exemplo, o nível de intensidade luminosa escolhido para uma divisão num dia nublado, pode ser conseguido com a simples abertura de cortinas, enquanto que num dia nublado, ou de noite, esse mesmo nível de iluminação poderá obrigar ao uso de luz artificial.

Situações problemáticas que devem ser encaradas com muita importância são os erros de previsão. Uma previsão errada pode ter as seguintes razões:

- Mudança de rotina: um habitante pode, excepcionalmente ou permanentemente, alterar a hora a que acorda ou que chega a casa, por exemplo. Esta alteração impede a previsão, por parte do sistema, das horas a que as novas actividades terão lugar. Imagine-se os seguintes cenários:

| Actividades do utilizador A | Interpretação do sistema, em fase de aprendizagem |
|--|--|
| 7h30 - Utilizador acorda e abre os estores | <ul style="list-style-type: none"> ▪ Treinar entrada da rede neuronal responsável por este equipamento com a configuração escolhida. |
| 7h31 - Liga o AC a 22°C | <ul style="list-style-type: none"> ▪ Treinar entrada da rede neuronal responsável por este equipamento com a configuração necessária para que às 7h30 a temperatura da divisão ser de 22°. |
| 7h40 - Utiliza água fria do lavatório | <ul style="list-style-type: none"> ▪ Este evento não tem influência no equipamento de aquecimento de água. |
| 7h50 - Toma um duche a 25°C | <ul style="list-style-type: none"> ▪ A utilização de água quente exige que o sistema de aquecimento de água, que está desligado para a redução do consumo de energia, seja activado com a antecedência necessária para que pouco antes das 7h50 o utilizador possa cumprir o seu desejo sem notar qualquer intervenção do sistema. Treinar entrada da rede neuronal responsável pelo equipamento de aquecimento de água de forma a este requisito ser satisfeito. |

Tabela 4.1 - Cenário A: Aprendizagem de uma rotina do utilizador A.

Numa fase inicial o módulo de inteligência não possui qualquer conhecimento sobre as rotinas do habitante. O cenário A indica a interpretação do módulo de inteligência das informações recolhidas durante a observação das actividades do ocupante.

Como as actividades não coincidem sempre com as rotinas aprendidas é necessário ser capaz de interpretar as novas acções e decidir se estas fazem parte de uma nova rotina ou se constituem apenas acções independentes.

O cenário B apresenta as alternativas possíveis na avaliação do carácter das novas acções.

| Actividades do utilizador A | Interpretação/Decisão do sistema | Alterações à rede |
|--|---|--|
| 7h00 - Utilizador acorda, acende a luz e abre os estores | <ul style="list-style-type: none"> Opção 1: Treinar entrada da rede neuronal responsável por este equipamento com a configuração escolhida, pois esta nova actividade pode ser permanente. | <ul style="list-style-type: none"> É necessário registar a nova alteração; É necessário eliminar a anterior (acender luz e abrir estores às 7h30); |
| | <ul style="list-style-type: none"> Opção 2: Interpretar esta actividade como um caso exceptional, que não deve ser registado. | <ul style="list-style-type: none"> Não há alterações. |
| 7h00 - Liga o AC a 22°C | <ul style="list-style-type: none"> Opção 1: Treinar entrada da rede neuronal responsável por este equipamento com a configuração necessária para que às 7h00 a temperatura da divisão ser de 22°. | <ul style="list-style-type: none"> É necessário registar a nova alteração; É necessário eliminar a anterior (Ligar AC às 7h30); |
| | <ul style="list-style-type: none"> Opção 2: Interpretar esta actividade como um caso exceptional, que não deve ser registado | <ul style="list-style-type: none"> Não há alterações. |
| 7h05 - Utiliza água fria do lavatório | Este evento não tem influência no equipamento de aquecimento de água. | N/A |
| 7h15 - Toma um duche a 25°C | <ul style="list-style-type: none"> Opção 1: Treinar entrada da rede neuronal responsável por este equipamento com a configuração necessária para que às 7h15 já seja possível utilizar tomar o duche com as condições desejadas. | <ul style="list-style-type: none"> É necessário registar a nova alteração; É necessário eliminar a anterior (abrir estores às 7h30. |
| | <ul style="list-style-type: none"> Opção 2: Interpretar esta actividade como um caso exceptional, que não deve ser registado | <ul style="list-style-type: none"> Não há alterações. |

Tabela 4.2 - Cenário B: Interpretação do sistema quando confrontado com um novo cenário.

Nestas situações a opção a tomar não é de fácil decisão. Pode ser mais apropriado aguardar para verificar se as novas actividades se vão tornar numa nova rotina, ou se as anteriores preferências ainda se mantêm. O recurso a uma base de dados para registar o historial de actividades pode ser uma solução. Assim torna-se possível verificar se um novo conjunto de actividades constituem a translacção no tempo de uma anterior rotina ou se representa ele próprio uma rotina extra.

- Erro de previsão da rede: Estas situações são extremamente indesejáveis, pois criam a noção de um sistema que controla abusivamente a nossa habitação. Estes erros podem ser resultado de uma conversão de valores pouco optimizada ou indicar que a rede ultrapassou a sua capacidade de armazenagem de informação, o que torna necessário modificar a estrutura da rede. Pode também indicar que a rede não se encontra optimizada, sendo necessária a alteração de um ou mais parâmetros para se conseguir atingir ou aproximar da configuração óptima global.

Nos próximos capítulos descreve-se em detalhe a implementação das aplicações desenvolvidas.

Capítulo 5

Implementação Rede Neuronal HomeSim

Neste capítulo descreve-se a implementação do módulo de inteligência da solução. São descritos os passos que levaram à decisão da estrutura da rede neuronal a ser implementada e da escolha dos seus parâmetros.

5.1 Arquitectura da rede

Foi considerada a escolha de uma rede baseada numa Função de Base Radial (FBR), devido principalmente ao facto deste tipo de rede convergir mais rapidamente do que uma RNA com funções sigmoidais. Mas, como também já foi provado, uma RNA com FBR exige, para se obter soluções equivalentes em termos de precisão, mais do que o triplo do número de neurónios escondidos, o que numa rede com um grande número de nós se torna impraticável em termos de requisitos computacionais e velocidade de resposta. Com este aumento do número de nós escondidos, numa RNA com FBR, o conjunto de amostras de treino tem também de ser bastante maior para se conseguirem bons resultados. Outra das vantagens de utilizar funções sigmoidais é o facto de um nó com esta função de activação conseguir responder a um intervalo de valores de entrada mais alargado, ao contrário de um nó com função de activação de base radial que responde apenas a uma região de valores reduzida. Assim, quanto maior for o número de nós de entrada e a sua gama de valores, mais nós escondidos são necessários. Por estas razões optou-se por utilizar uma RNA com função de activação sigmoideal. Como os resultados dependem fortemente da arquitectura escolhida para a rede, nomeadamente do número de camadas e de

nós, do tipo de função de activação e do tipo de algoritmo de treino, são descritos em seguida os passos seguidos na optimização da solução.

5.1.1 Escolha do número de camadas

Foi já provado que uma rede neuronal constituída por três camadas e com um número ilimitado de nós na camada escondida consegue classificar qualquer função não-linear, sendo a configuração mais simples que possui esta capacidade. Assim, a escolha recaiu numa rede com uma camada de entrada, uma camada escondida e uma camada de saída.

5.1.2 Escolha do número de nós das várias camadas

A escolha do número de entradas da rede é sempre um compromisso entre a quantidade de informação relacionada e o processamento computacional necessário para o tratamento desta mesma informação. Utilizar uma só rede neuronal artificial para registar e prever a informação relativa a todas as actividades de uma casa obrigava a que o número de nós escondidos fosse muito elevado, para conseguirem organizar-se de forma a capturar os traços que permitem uma previsão acertada. Forçar uma RNA com um número de nós inferior ao adequado a registar um número elevado de actividades leva à perda da informação armazenada anteriormente, pois é ultrapassada a capacidade associativa da rede. Assim foi criada uma rede neuronal por cada equipamento a gerir, conseguindo assim uma capacidade apurada para replicar as acções aprendidas

Por forma a permitir um ambiente multi-utilizador, mas ao mesmo tempo com preocupação em evitar a sobrecarga de informação armazenada na rede foi decidido contemplar a presença de até quatro utilizadores na habitação, representados pelo número de aplicações cliente activas simultaneamente.

É fundamental registar a informação temporal das actividades como dia da semana, hora e minuto. Optou-se por não se fazer o registo dos meses, pela pouca importância que acresce em termos de teste, bem como dos segundos em que as acções tiveram lugar.

É igualmente importante contemplar as condições ambientais de intensidade luminosa, temperatura, humidade e fumo, pela influência directa que têm nas acções tomadas pelos utilizadores. A informação relativa a estas condições é captada pela presença de detectores nas várias divisões da habitação. Considerando a diferente gama de valores possíveis, dada a natureza variável que mesmo uma rotina possui, é necessário fazer o correcto manuseamento dos valores de entrada para que a rede se adapte com resultados satisfatórios. Assim, apresenta-se em seguida a lógica que levou à decisão do número de nós de entrada na rede, sabendo que uma

maior independência entre os valores de entrada otimiza a capacidade da rede de distinguir as amostras aprendidas:

- Utilizador:

Considerando que as acções de um utilizador quando se encontra sozinho numa sala são diferentes daquelas que toma quando partilha o espaço com um ou mais utilizadores, é necessário considerar todas as combinações possíveis de utilizadores num espaço. Assim, havendo 4 utilizadores, existem as seguintes possibilidades:

- Todos os utilizadores sozinhos - 4 casos;
- Dois utilizadores a partilhar o espaço - $C_2^4 = \frac{4!}{2!(4-2)!} = 6$ casos;
- Três utilizadores a partilhar o espaço - $C_3^4 = \frac{4!}{3!(4-3)!} = 4$ casos;
- Quatro utilizadores a partilhar o espaço - 1 caso;

No total há 15 combinações possíveis. Se fosse apenas utilizado um nó de entrada para representar estas diferentes combinações, os diferentes valores estariam separados por $\frac{1}{15-1} \cong 0,071$. Se a escolha recaísse sobre dois nós, e codificando igualmente de forma linear, os diferentes valores de entrada estariam separados por $\frac{2}{15-1} \cong 0,142$. Utilizando quatro nós seria possível usar uma lógica binária para representar a presença dos diferentes utilizadores numa divisão, e sairia reforçada baixa correlação entre as actividades dos diferentes utilizadores. Foram reservados quatro nós para indicar a presença dos utilizadores na divisão em que decorreu a acção;

- Dia:

Para efeitos de teste será considerado apenas os dias da semana e não os dias do mês, pois é com os primeiros que as rotinas variam principalmente. Assim, os valores de entrada relativos aos sete dias da semana estão separados, utilizando apenas um nó de entrada, por $\frac{1}{7-1} \cong 0,167$, o que permite uma considerável independência entre diferentes valores;

- Hora:

Havendo 24 horas num dia, se apenas fosse utilizada uma entrada para expressar os diferentes valores eles estariam separados por $\frac{1}{24-1} \cong 0,043$. No caso de se utilizar dois nós essa mesma separação seria de $\frac{2}{24-1} \cong 0,086$. Optou-se por esta última possibilidade pois permite sectorizar diferentes alturas do dia de forma adequada;

- Minuto:

Neste caso há a necessidade de diferenciar 60 valores de entrada. Utilizando apenas um nó os diferentes valores estariam muito próximos, pelo que não se coloca essa hipótese. Utilizando dois nós de entrada os valores estariam separados por $\frac{2}{60-1} \cong 0,033$, e com três a diferença entre valores seria igual a $\frac{3}{60-1} \cong 0,05$. Apesar de ser um valor pequeno este não é o único factor a considerar, pois como será explicitado a utilização de três nós permite uma boa diferenciação entre diferentes intervalos de tempo;

- ID do objecto alvo da última acção:

A natureza sequencial das nossas actividades leva a que seja considerado este parâmetro. Se para indicar o valor de identificação dos 23 dispositivos recriados no simulador se utilizar um nó de entrada a diferença entre valores consecutivos será de $\frac{1}{23-1} \cong 0,05$. Com um apropriado entrelaçamento destes valores a correlação entre diferentes entradas é reduzida;

- Condições ambientais:

A necessidade de maximizar a diferença entre entradas, minimizando o número de nós ocupados, atinge principal relevo quando a entrada tem uma gama de 101 valores diferentes (0-100).

Em termos de intensidade luminosa, humidade e fumo, pequenas diferenças entre valores não se traduzem numa alteração significativa das condições perceptíveis, pelo que a utilização de um nó de entrada, mesmo resultando em valores de entrada separados por $\frac{1}{101-1} \cong 0,001$, foi a escolhida, com a possibilidade de este número ser alterado no caso destes dados serem incorrectamente apresentados

Em relação aos nós de saída a situação é idêntica à avaliação do número de nós para representar as condições ambientais. Uma gama de 101 valores (0-100) representa todos os valores possíveis das saídas dos vários equipamentos. Porém, nesta situação é necessário ter presente que os valores de saída de uma rede são afectados por erros e por *jitter*, ao contrário dos valores de entrada que são apresentados com a precisão desejada. Assim torna-se necessária a utilização de dois nós de saída, que se traduz numa separação entre valores igual a $\frac{2}{101-1} \cong 0,002$.

Foram escolhidos o seguinte número de nós para as diferentes camadas:

- Nós de entrada - 15;
- Nós da camada escondida - $2 \times n^{\circ} \text{ de nós de entrada} + 1 = 31$;
- Nós de saída - 2.

5.1.3 Treino da rede

Foi utilizado o algoritmo de retropropagação. Para evitar o excesso de treino adoptou-se uma técnica de paragem antecipada. Para adoptar este processo o conjunto de amostras criado foi dividido em três subconjuntos:

- Subconjunto de treino - composto por 60% do conjunto total. Utilizado para o ajuste dos pesos das ligações;
- Subconjunto de validação - composto por 30% do conjunto total. Utilizado para reduzir o erro de validação;
- Subconjunto de teste - composto pelos restantes 10%. Utilizado para testar a solução obtida.

5.1.4 Pârametros de ajuste

Utilizaram-se os seguintes valores iniciais para os diferentes parâmetros:

- Passo de aprendizagem: $\eta = 0,1$;
- Momento: $\alpha = 0,01$;
- *Jitter* máximo: 0,003;
- Ciclos de treino: 150.

O valor escolhido para o passo de aprendizagem é baixo, para minimizar a probabilidade de a solução ficar retida num mínimo local. Um valor baixo traduz-se, porém, numa lenta convergência da solução. O valor escolhido para o momento é também baixo, mas serve o propósito de acelerar um pouco a convergência da rede e ajudar a solução a transpôr mínimos locais. Um momento elevado pode impedir a rede de encontrar o óptimo global ou mesmo um bom óptimo local. O *jitter* é responsável pela criação de ruído, que afecta os resultados da rede. O número de ciclos de treino é um dos parâmetros cujo ajuste será alvo de teste.

5.2 Implementação

De forma a manter as aplicações clientes sempre actualizadas, para além de fazer a gestão das condições ambientais, as alterações no estado da casa induzidas por um utilizador são enviadas às restantes aplicações clientes activas de forma a manter o seu estado sempre actualizado. Assim consegue-se que o cenário visualizado seja comum a todos os utilizadores, recriando da melhor forma o conceito de espaço partilhado.

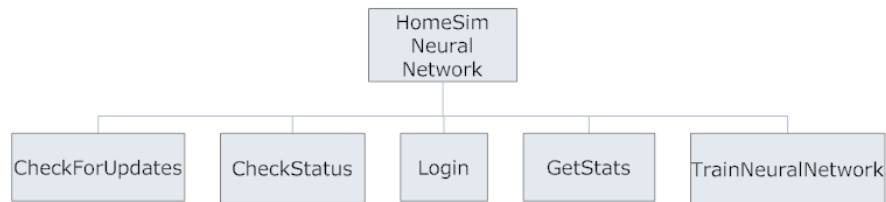


Figura 5.1 - Funções implementadas pela aplicação Rede Neuronal HomeSim.

Esta aplicação, onde estão implementadas as diversas redes neuronais necessárias, faz a gestão das condições da habitação, partilhadas por todos. A RNH disponibiliza um *web service*, que permite o acesso às seguintes funções:

- *CheckForUpdates*:
Função invocada para saber se houve alteração no estado de algum dispositivo ou na localização da personagem;
- *CheckStatus*:
Esta função é invocada quando um novo utilizador acede ao HomeSim, para tomar conhecimento das personagens que estão a ser utilizadas por outros utilizadores e da sua localização na casa;
- *GetStats*:
Esta função retorna os valores associados aos nós da rede neuronal indicada. É utilizada pela Rede Neuronal Visual;
- *Login*:
Indica à Rede Neuronal HomeSim qual das personagens livres o utilizador pretende utilizar para simular as actividades na casa;
- *TrainNeuralNetwork*:
Envia à Rede Neuronal HomeSim a passagem de uma personagem para uma nova divisão da casa ou a alteração do estado de um objecto;

Foi estudado qual o método mais eficiente em termos de rapidez de aprendizagem e em qualidade dos resultados obtidos. Os estímulos/tarefas podem ser fornecidos à rede neuronal de forma sucessiva, que é a abordagem mais natural devido ao carácter cronológico das tarefas, ou de forma simultânea, em que são agrupados de acordo com a tarefa associada e fornecidos discriminadamente. Não foi encontrada uma diferença significativa entre as duas abordagens.

5.2.1 A plataforma NeuronDotNet

Para a implementação da rede neuronal foi utilizada a plataforma NeuroDotNet. Esta é uma plataforma, desenvolvida em C#, que apesar de não ser tão poderosa como as disponíveis para outras linguagens, como o MatLab por exemplo, permite a criação de redes neuronais de retropropagação com complexidade suficiente para a implementação desejada. Sendo distribuída segundo as licenças GPL e LGPL torna-se possível a alteração do código fonte para a optimização dos resultados da solução.

5.2.2 Implementação das rede neuronais

Foi criada uma RNA por equipamento a controlar e uma rede para a representação das condições de cada divisão. No total foram $23 + 6 = 29$ redes que actuam juntamente para que a casa se comporte como um ser inteligente.

5.3 Rede Neuronal Visual

A aplicação Rede Neuronal Visual (RNV) foi desenvolvida para a análise dos parâmetros da rede, de forma a facilitar a interpretação do funcionamento da rede e a sua optimização. Esta aplicação implementa as seguintes funções:

- *CheckForUpdate*:
Verifica se a rede observada sofreu alterações. Se tal tiver acontecido é retornado um novo conjunto de valores e é invocada a função *UpdateTable*;
- *UpdateTable*:
Recebe um conjunto de valores que representa os vários parâmetros que caracterizam os nós de uma rede e actualiza os valores apresentados.

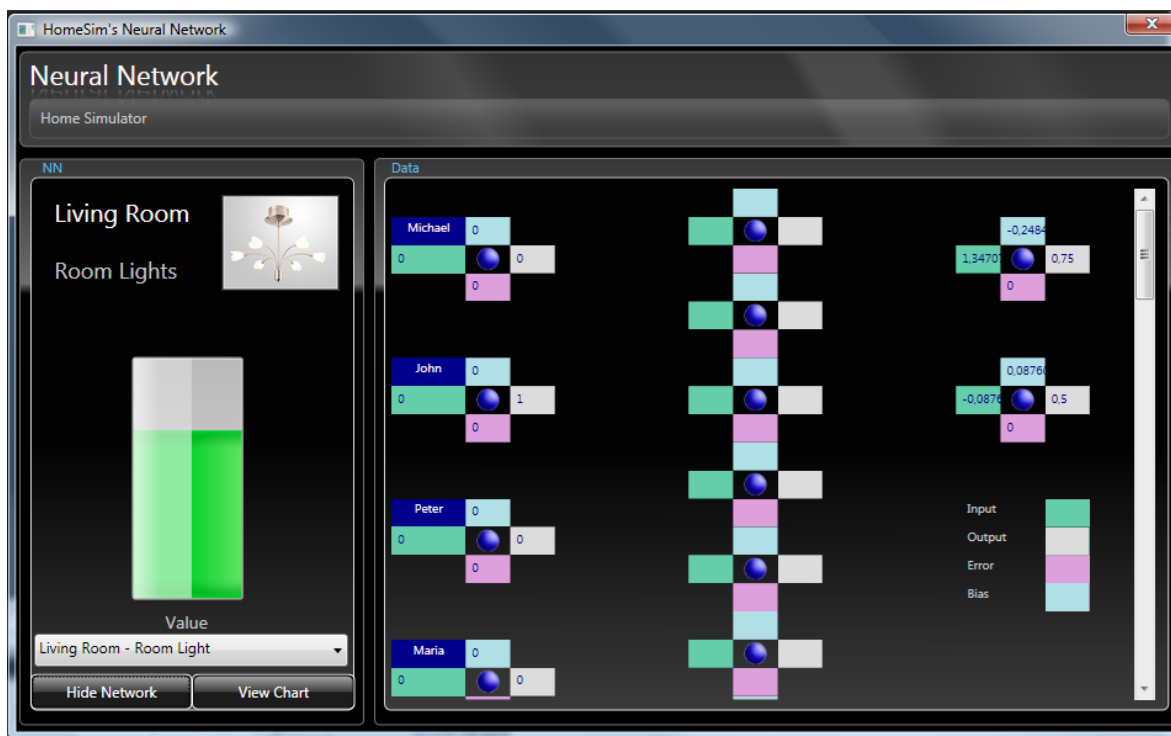


Figura 5.2 - Interface da aplicação Rede Neuronal Visual.

5.4 Resumo

A diferença entre utilizar o simulador e um cenário de teste real, que poderia ser implementado recorrendo ao sistema de controlo previamente desenvolvido, prende-se com o tratamento mais simples das mensagens. No cenário real seria necessário processar a mensagem ZigBee, identificar o equipamento remetente da mensagem através do endereço MAC de origem e fazer a conversão dos dados para o formato desejado consoante o tipo de dispositivo anunciado. Com o simulador tira-se partido da comunicação por *web services* com a Rede Neuronal HomeSim, o que permite ultrapassar as dificuldades inerentes à utilização de *hardware* de teste e focar no desenvolvimento da solução pretendida.

Capítulo 6

Implementação Aplicação HomeSim

Esta aplicação foi criada com o propósito de simular uma habitação. Nela podem ser recriadas algumas das acções comuns num contexto habitacional como ligar/desligar luzes, ar condicionado, máquinas de lavar, entre outras acções. Para que a aplicação pudesse ser testada por vários utilizadores de modo fácil e assim contribuir para o seu desenvolvimento, a solução ideal era aceder ao HomeSim através da *internet*. Este tipo de acesso obriga a ter preocupação com a compatibilidade e tamanho da aplicação.

6.1 Implementação

Para evitar que o utilizador tivesse de descarregar a própria aplicação para o seu computador, perdendo-se assim a facilidade de acesso, decidiu-se apostar numa tecnologia recente, cujas ferramentas de desenvolvimento estão ainda elas próprias a ser desenvolvidas: o Windows Presentation Foundation (WPF). O tipo de aplicação WPF adequado à solução a implementar é o XAML *Browser Application* (XBAP), que permite que qualquer computador que tenha a .NET Framework 3.0 instalada aceda a uma aplicação da mesma forma que se acede a uma página *web*, não necessitando o utilizador de qualquer tipo de intervenção extra. As aplicações XBAP têm algumas limitações relativamente a uma aplicação WPF convencional devido a correrem num *browser*, como o facto de o utilizador ter de estar *online* para aceder à aplicação. Os *browsers* actuam com fortes medidas de segurança, o que por si só também torna mais difícil desenvolver uma aplicação para este ambiente. Com a convicção de que esta era a escolha acertada decidiu-se implementar o simulador com base nesta tecnologia.

Como o HomeSim corre na máquina de cada utilizador, a aplicação tem de comunicar com o módulo de inteligência de uma forma segura e fiável. O XBAP só permite que a aplicação comunique com a máquina a partir de qual a aplicação foi acedida. Ambas as aplicações, o HomeSim e o Rede Neuronal HomeSim, foram desenvolvidas na mesma máquina, por uma questão de conveniência.

Esta aplicação implementa as seguintes funções:

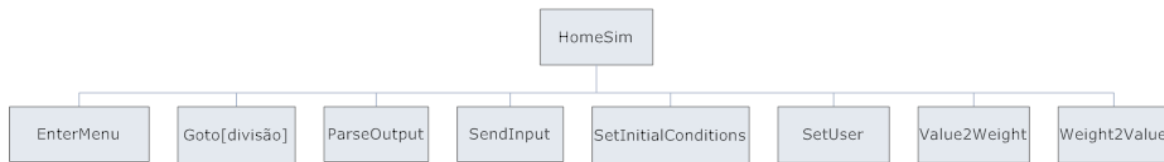


Figura 6.1 - Funções implementadas pelo HomeSim.

Descreve-se em seguida o papel de cada uma das funções:

- *EnterMenu:*
Pede à Rede Neuronal HomeSim o estado e localização de todas as personagens. Assim garante-se que uma personagem é apenas utilizada simultaneamente por um utilizador;
- *Goto[Nome da divisão]:*
Esta função é invocada quando um utilizador pretende transitar para outra divisão da habitação. É feita a gestão da mudança em termos de interface e comunicada a nova localização da personagem ao servidor;
- *ParseOutput:*
Como uma aplicação XBAP não suporta a utilização de *web services* em modo *full-duplex*, em que qualquer dos terminais pode iniciar a comunicação, o HomeSim pergunta repetidamente, com um intervalo de tempo definido, se existe alguma nova previsão. Em resposta a uma destas perguntas, quando à Rede Neuronal HomeSim prevê uma actividade, é enviada a informação do novo estado do equipamento que deve ser actualizado;
- *SendInput:*
Esta função é invocada quando ocorreu alguma acção que deve ser comunicada à Rede Neuronal HomeSim, como a alteração do estado de um dispositivo;
- *SetInitialConditions:*
Pede à Rede Neuronal HomeSim o estado de todos os equipamentos da habitação e as condições de cada divisão, de forma a actualizar o simulador;
- *SetUser:*
Comunica ao servidor qual a personagem escolhida para efectuar a simulação;

- *Value2Weight*:
Faz a codificação dos valores do estado dos equipamentos;
- *Weight2Value*:
Faz a decodificação dos valores do estado dos equipamentos.

6.1.1 Codificação dos valores de entrada

Antes de enviar dados para o HomeSimNN é necessário fazer a sua conversão para a gama de valores adequada. Para tal é necessário codificar os valores de saída das acções pretendidas de forma a poderem ser utilizados como entradas da rede neuronal.

A codificação dos valores tem uma importância fulcral no bom desempenho da solução, pois ela dita o aproveitamento da grande vantagem da utilização de uma rede neuronal: a capacidade de generalização.

Em seguida apresenta-se a lógica por detrás da codificação implementada para cada tipo de valores:

- *Utilizador*:
Utilizando quatro nós para representar a presença de cada utilizador na divisão em que o objecto foi actuado permite utilizar uma lógica binária. Assim, as entradas respectivas aos utilizadores que estiverem na divisão no momento de interacção com o objecto são têm valor '1' e as restantes '0'. Não é necessário indicar o autor da alteração, pois o importante é o ponto de vista da habitação. O importante é saber que quando os utilizadores A e B estão presentes numa divisão, a intensidade luminosa da sala, por exemplo, é colocada a 75%, não sendo necessário identificar o responsável por essa alteração;
- *Dia*:
Nesta caso a codificação linear dos valores é suficiente, pelo que a cada dia da semana corresponde o valor presente no vector [0, 0.16, 0.33, 0.5, 0.67, 0.83, 1];

- Hora:

Para tirar partido da capacidade de generalização das redes neuronais, utilizando duas entradas para a representação dos valores, é necessário sectorizar as 24 horas de um dia de forma óptima. Sabendo que as horas de pico de actividade numa habitação são entre as 7h e as 9h, entre as 12h30 e a 14h e entre as 19h e as 23h, optou-se por utilizar os valores de uma das entradas para fazer a separação entre quatro períodos distintos do dia, e a outra entrada para precisar qual a hora em que a actividade teve lugar. A divisão considerada óptima foi a seguinte:

| Hora | Entrada 1 | Entrada 2 |
|------|-----------|-----------|
| 1 | 0 | 0 |
| 2 | 0 | 0,2 |
| 3 | 0 | 0,4 |
| 4 | 0 | 0,6 |
| 5 | 0 | 0,8 |
| 6 | 0 | 1 |
| 7 | 0,33 | 0 |
| 8 | 0,33 | 0,2 |
| 9 | 0,33 | 0,4 |
| 10 | 0,33 | 0,6 |
| 11 | 0,33 | 0,8 |
| 12 | 0,33 | 1 |
| 13 | 0,66 | 0 |
| 14 | 0,66 | 0,2 |
| 15 | 0,66 | 0,4 |
| 16 | 0,66 | 0,6 |
| 17 | 0,66 | 0,8 |
| 18 | 0,66 | 1 |
| 19 | 1 | 0 |
| 20 | 1 | 0,2 |
| 21 | 1 | 0,4 |
| 22 | 1 | 0,6 |
| 23 | 1 | 0,8 |
| 24 | 1 | 1 |

Tabela 6.1 - Codificação das horas em valores aceites por uma rede neuronal.

Utilizando esta divisão das horas consegue-se um maior grau de correlação entre os valores atribuídos a horas consecutivas. Sabendo que mesmo as actividades de rotina não são executadas sempre no mesmo exacto momento, esta superior correlação permite à rede associar uma actividade realizada às 7h45 com uma que teve lugar às 8h05, por exemplo, pois o valor de um dos nós é o mesmo e o valor do segundo nó apenas varia 0,2;

- Minuto:

Neste caso há a necessidade de codificar 60 valores diferentes, o que se torna uma tarefa mais complicada. A abordagem passa por compreender a natureza humana de, desde que nos acompanhamos por um relógio, efectuar acções em alturas bem definidas. Assim, existe uma maior probabilidade de, intencionalmente, iniciar uma actividade num período em torno dos 0, 15, 30 ou 45 minutos. Um dos exemplos é o facto de ligarmos a televisão para observarmos um determinado programa, que tem início geralmente nos minutos indicados. Assim, a divisão de valores de uma das entradas (entrada 1) foi feita em torno dos valores indicados. A codificação entendida como óptima foi a seguinte:

| Minuto | Entrada 1 | Entrada 2 | Entrada 3 |
|--------|-----------|-----------|-----------|
| 53 | 0 | 0 | 0,5 |
| 54 | 0 | 0,25 | 1 |
| 55 | 0 | 0,5 | 0 |
| 56 | 0 | 0,75 | 0,5 |
| 57 | 0 | 1 | 1 |
| 58 | 0 | 0 | 0 |
| 59 | 0 | 0,25 | 0,5 |
| 0 | 0 | 0,5 | 1 |
| 1 | 0 | 0,75 | 0 |
| 2 | 0 | 1 | 0,5 |
| 3 | 0 | 0 | 1 |
| 4 | 0 | 0,25 | 0 |
| 5 | 0 | 0,5 | 0,5 |
| 6 | 0 | 0,75 | 1 |
| 7 | 0 | 1 | 0 |
| 8 | 0,33 | 0 | 0,5 |
| 9 | 0,33 | 0,25 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 21 | 0,33 | 0,75 | 1 |
| 22 | 0,33 | 1 | 0 |
| 23 | 0,66 | 0 | 0,5 |
| 24 | 0,66 | 0,25 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 36 | 0,66 | 0,75 | 1 |
| 37 | 0,66 | 1 | 0 |
| 38 | 1 | 0 | 0,5 |
| 39 | 1 | 0,25 | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 51 | 1 | 0,75 | 1 |
| 52 | 1 | 1 | 0 |

Tabela 6.2 - Codificação dos minutos.

- ID do objecto alvo da última acção:

O tipo de dispositivo que se repete por todas as divisões é de iluminação. Assim, há seis tipos iguais de objectos, pelo que se chegou a conclusão que atribuir um valor diferente a cada um não seria a melhor opção, devendo-se maximizar as diferenças entre valores procurando minimizar a correlação entre valores atribuídos a objectos comuns. A solução escolhida foi a seguinte:

| Divisão | Objecto | ID | Entrada 3 |
|-----------------|----------------|----|-----------|
| Cozinha | Luz | 11 | 0 |
| | Exaustor | 12 | 0,2 |
| | Máq. Loiça | 13 | 0,6 |
| | Estores | 14 | 1 |
| Sala de Estar | Luz | 21 | 0,4 |
| | Estores | 22 | 0,8 |
| | TV | 23 | 0 |
| | Colunas | 24 | 0,2 |
| Sala de Jantar | Luz | 31 | 0,8 |
| | Estores | 32 | 0,4 |
| | TV | 33 | 0,6 |
| | Colunas | 34 | 0,8 |
| Quarto | Luz | 41 | 0,2 |
| | Estores | 42 | 0,2 |
| | AC | 43 | 1 |
| Escritório | Luz | 51 | 0,6 |
| | AC | 52 | 0,8 |
| | Estores | 53 | 0 |
| | TV | 54 | 1 |
| | Radio | 55 | 0,2 |
| Quarto de banho | Luz principal | 61 | 1 |
| | Luz do espelho | 62 | 0,4 |
| | Banheira | 63 | 0,8 |
| | Lavatório | 64 | 0,2 |

Tabela 6.3 - Codificação do número de identificação dos equipamentos da habitação.

- Condições ambientais:

Como já foi referido, em termos de intensidade luminosa, humidade e concentração de fumo, pequenas diferenças entre valores são distinguidas pelos sensores mas menos perceptíveis ao ser humano. Assim optou-se por uma codificação linear do valor: [0.00, 0.01, ..., 0.99, 1]. Porém, uma diferença de temperatura já é mais perceptível, pelo que é possível utilizar a gama de valores de forma mais adequada. Considerando que a

temperatura numa habitação (e da água utilizada no lavatório e banheira) varia entre os 5° e 30°:

| Temperatura (°C) | Entrada |
|---------------------|---------|
| 5 | 0 |
| 6 | 0,04 |
| 7 | 0,08 |
| ⋮ | ⋮ |
| 28 | 0,92 |
| 29 | 0,96 |
| 30 | 1 |

Tabela 6.4 - Codificação dos valores de temperatura.

6.1.2 Codificação dos valores de saída

Como já referido, os valores de saída de uma rede são afectados por erros e por *jitter*, ao contrário dos valores de entrada. Assim, em vez da codificação linear de valores, opta-se por uma conversão que minimize o erro. A gama de valores de saída varia conforme o tipo de dispositivo. Um equipamento de iluminação consegue ter uma gama de 100 valores diferentes, se classificarmos o nível de iluminação como um valor percentual detectado por um sensor de luz. Já para um estore não faz sentido haver uma gama de 100 valores diferentes, até porque o motor que o actua pode não permitir tal precisão. Para um estore 0% de luminosidade indica o seu fecho total e 100% a posição oblíqua que permite a melhor luminosidade natural. Esta foi a codificação idealizada:

- Intensidade luminosa / volume:

Uma pequena diferença de intensidade luminosa ou volume é pouco perceptível, excepto quando essa diferença representa o estado activo ou inactivo do equipamento de luz. Devido ao erro de saída da rede neuronal ser passível de ser reduzido mas impossível de eliminar, é necessário criar uma margem de segurança que assegure o estado inactivo do equipamento. Quanto ao nível de intensidade luminosa, o valor enviado para a rede neuronal, como amostra de treino, não é o valor de regulação da luz artificial mas sim o valor registado pelo sensor de luz da divisão após a alteração do estado do dispositivo. Assim é registada a preferência do utilizador, que será satisfeita sempre que possível recorrendo à luz natural, através da abertura de estores.

Considerando que com uma variação de 5% se garantem todos os níveis de intensidade desejados:

| Intensidade luminosa / volume (%) | Saída 1 | Saída 2 |
|---|---------|---------|
| 0 | 0 | 0 |
| 5 | 0,2 | 0,25 |
| 10 | 0,2 | 0,5 |
| 15 | 0,2 | 0,75 |
| 20 | 0,2 | 1 |
| 25 | 0,4 | 0 |
| 30 | 0,4 | 0,25 |
| 35 | 0,4 | 0,5 |
| 40 | 0,4 | 0,75 |
| 45 | 0,4 | 1 |
| 50 | 0,6 | 0 |
| 55 | 0,6 | 0,25 |
| 60 | 0,6 | 0,5 |
| 65 | 0,6 | 0,75 |
| 70 | 0,6 | 1 |
| 75 | 0,8 | 0 |
| 80 | 0,8 | 0,25 |
| 85 | 0,8 | 0,5 |
| 90 | 0,8 | 0,75 |
| 95 | 0,8 | 1 |
| 100 | 1 | 1 |

Tabela 6.5 - Codificação da intensidade luminosa de saída.

- Temperatura AC/Água:

Como a diferença de temperatura é mais perceptível procurou-se eliminar ao máximo a capacidade de uma saída afectada de erro representar um novo valor de temperatura. Estes foram os valores encontrados:

| Temperatura (°C) | Saída 1 | Saída 2 |
|---------------------|---------|---------|
| 5 | 0 | 0 |
| 6 | 0,04 | 0,04 |
| 7 | 0,08 | 0,08 |
| 8 | 0,12 | 0,12 |
| 9 | 0,16 | 0,16 |
| ⋮ | ⋮ | ⋮ |
| 26 | 0,84 | 0,84 |
| 27 | 0,88 | 0,88 |
| 28 | 0,92 | 0,92 |
| 29 | 0,96 | 0,96 |
| 30 | 1 | 1 |

Tabela 6.6 - Codificação da temperatura de saída.

- Rádio/TV:

Considerando 25 bandas/canais diferentes:

| Banda/Canal | Saída 1 | Saída 2 |
|-------------|---------|---------|
| Desligado | 0 | 0 |
| 1 | 0,04 | 0,04 |
| 2 | 0,08 | 0,08 |
| 3 | 0,12 | 0,12 |
| 4 | 0,16 | 0,16 |
| ⋮ | ⋮ | ⋮ |
| 21 | 0,84 | 0,84 |
| 22 | 0,88 | 0,88 |
| 23 | 0,92 | 0,92 |
| 24 | 0,96 | 0,96 |
| 25 | 1 | 1 |

Tabela 6.7 - Codificação da banda/canal de saída.

6.2 Resumo

Esta aplicação permite a um utilizador recriar as actividades mais comuns numa habitação. Na implementação do HomeSim teve-se a preocupação de criar a sensação de espaço partilhado. Isso é conseguido através da actualização permanente dos interfaces das aplicações simulador e do conhecimento das personagens que estão na mesma divisão que a personagem por nós controlada. A conversão de valores da sua gama real para o formato passível de ser interpretado por uma rede é feito nesta aplicação, libertando o módulo de inteligência para o processamento da rede neuronal e tomando partido do processamento distribuído por cada aplicação cliente. O interface do simulador foi alvo de particular atenção, pois torna-se bastante mais agradável e realista realizar testes num cenário perceptível e verídico do que numa aplicação simplista.

Capítulo 7

Avaliação do Trabalho

Neste capítulo é indicado o cenário de teste escolhido e são analisados os resultados das várias fases de desenvolvimento da solução.

7.1 Cenário de teste

O testes realizados centraram-se em torno de um cenário que contempla a utilização, por parte de um utilizador de vários recursos de uma habitação, ao longo de um dia. Usando este cenário é possível verificar o desempenho do módulo de inteligência, nas diferentes fases do desenvolvimento da solução. Este cenário contempla a utilização de vários objectos de várias divisões. Cada uma destas actividades foi treinada 150 vezes para que os traços de informação das entradas fossem capturados. Um valor mais reduzido traduzir-se-ia num tempo de treino inferior mas, provavelmente, em piores resultados. Atendendo à actividade de cada equipamento, sempre que havia um intervalo superior a 10 min entre duas acções pre-determinadas consecutivas, a rede foi treinada com um valor igual ao esperado nesse intervalo.

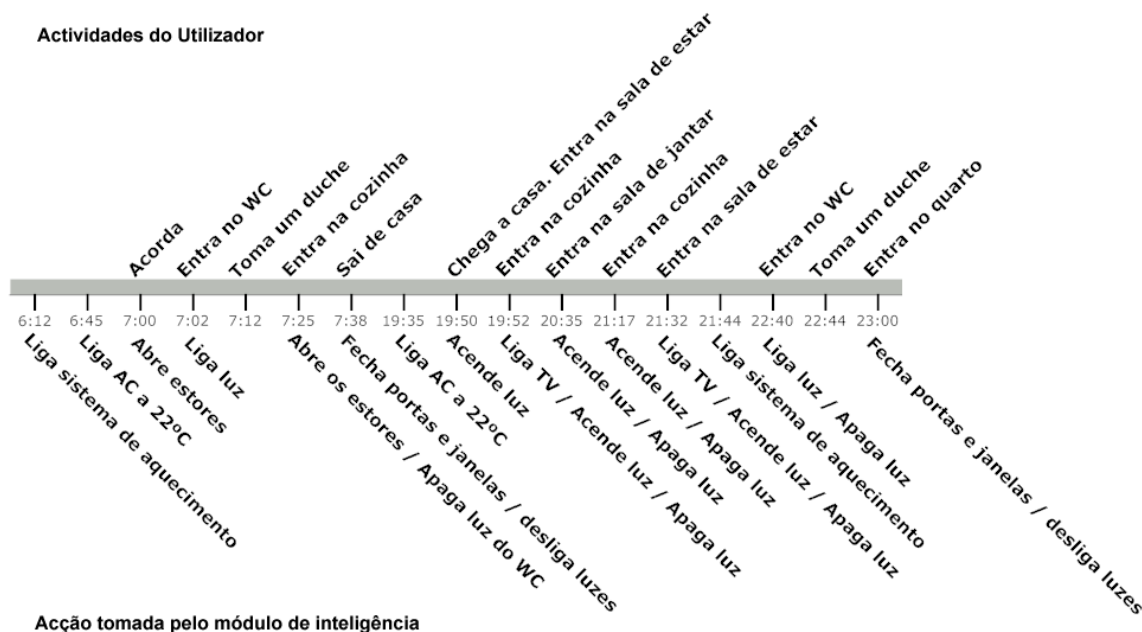


Figura 7.1 - Cenário de teste.

7.2 Testes realizados

Foram testadas três diferentes soluções. Utilizou-se inicialmente uma configuração cautelosa da rede e posteriormente foram tomadas medidas para otimizar o seu tempo de resposta. Os valores apresentados nos gráficos foram apenas calculados nos instantes em que actividades tiveram início e fim de cada actividade, sendo os pontos interligados por rectas.

7.2.1 Teste da rede com 31 nós escondidos

A solução inicial utiliza a abordagem mais cuidadosa no que diz respeito ao número de nós escondidos utilizados. A utilização de 31 ($2 \times n^{\circ} \text{ de nós de entrada} + 1$) nós escondidos é a que teoricamente permite obter os melhores resultados, apesar de ser a que exige maior poder de processamento no treino e na previsão de actividades.

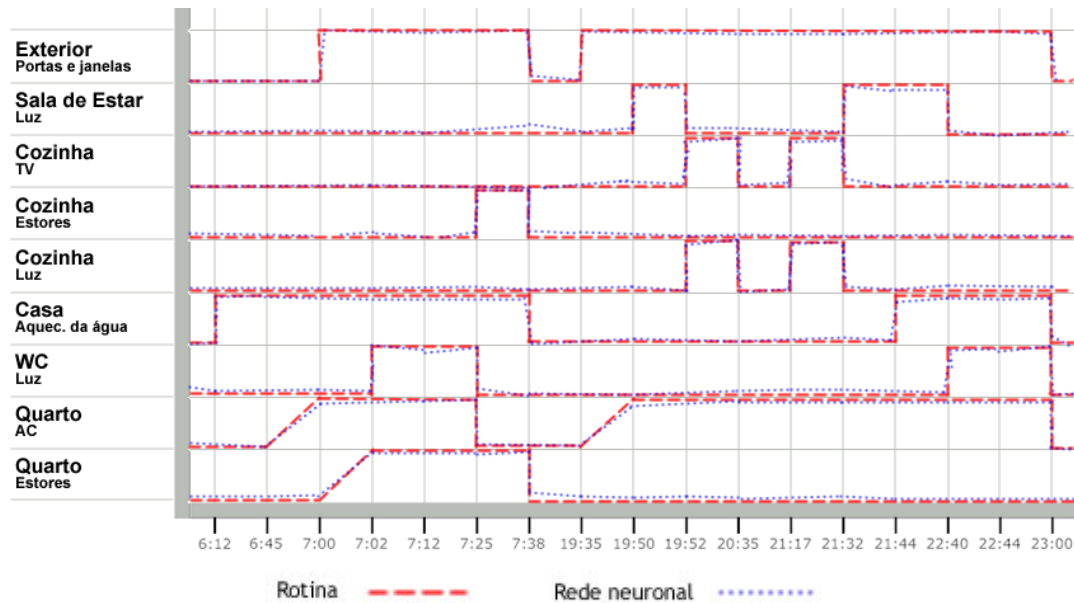


Figura 7.2 - Teste da rede com 31 nós escondidos.

7.2.2 Teste da rede com 13 nós escondidos

Neste teste para não sobrecarregar a rede com demasiada informação, apenas se o intervalo entre duas acções fosse superior a 20 min é que seria treinado o valor esperado nesse intervalo. Com o objectivo de aumentar a velocidade de resposta da rede e exigir menos poder de processamento foi-se progressivamente reduzindo o número de nós escondidos, até se encontrar um valor que traduzia um melhor balanço entre precisão de resultados e velocidade de resposta.

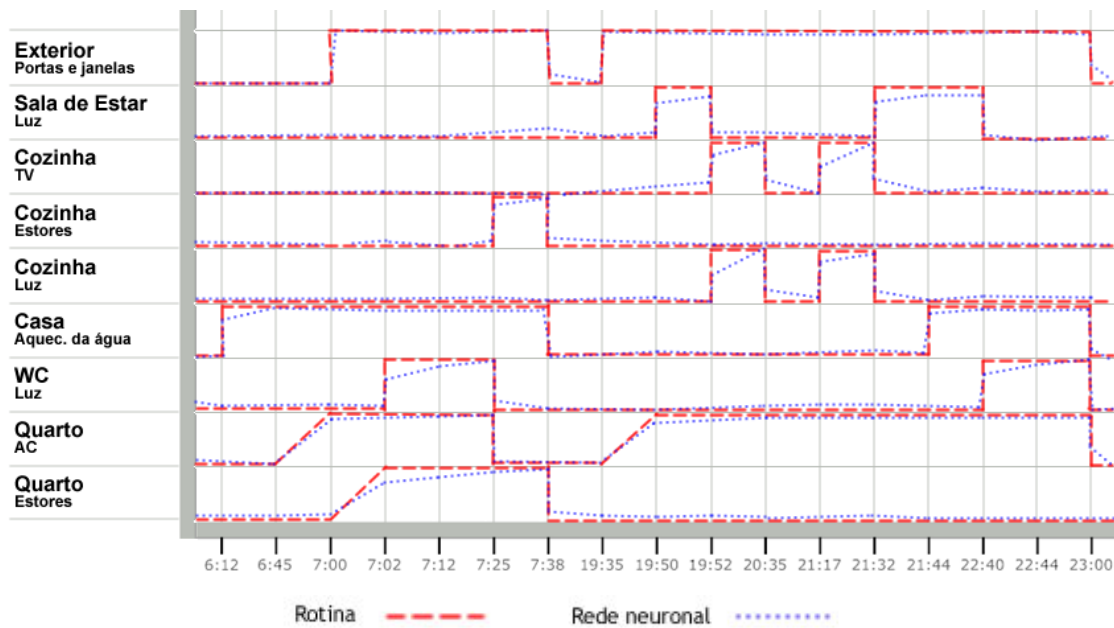


Figura 7.3 - Teste da rede com 13 nós escondidos.

7.2.3 Teste com alteração de parâmetros da rede

Para este teste, com o intuito de se tentar aproximar mais rapidamente da solução ótima de configuração dos pesos das ligações da rede, os valores do passo de aprendizagem e do momento foram ajustados até terem os valores de 0.2 e 0.03, respectivamente, com os quais foram conseguidos os melhores resultados para esta configuração. Mantiveram-se os 13 nós escondidos usados no último teste.

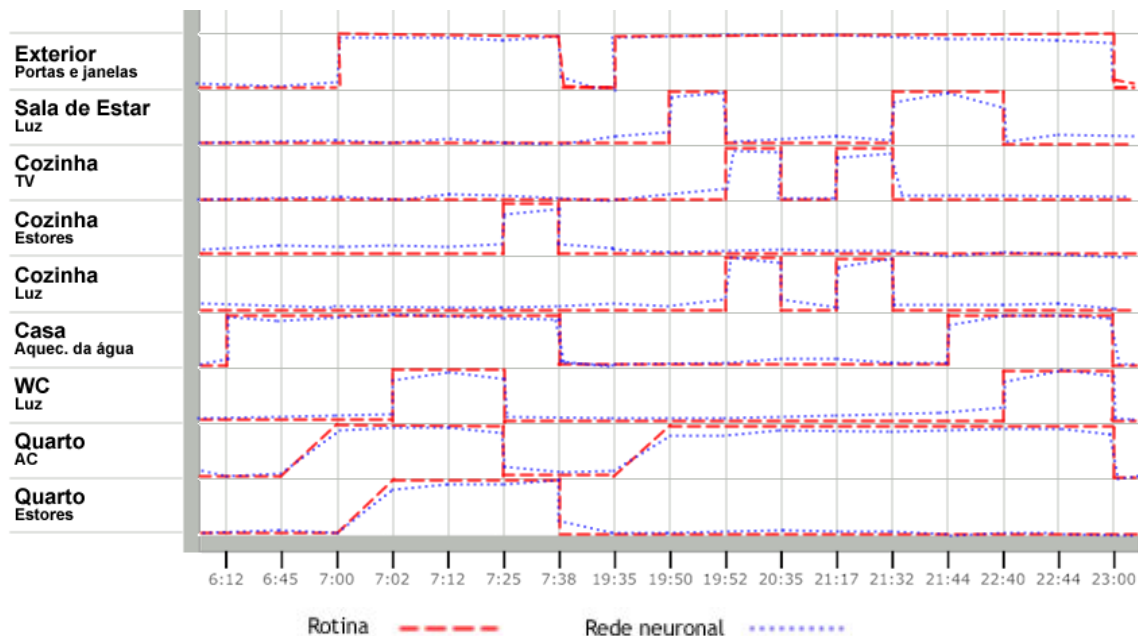


Figura 7.4 - Teste da rede com 13 nós escondidos, passo de aprendizagem 0.2 e momento 0.03.

7.2.4 Teste do número de intervenções do utilizador

Uma medida da qualidade dos resultados obtidos, com as diferentes arquitecturas, é o número de vezes em que o habitante teve de intervir e alterar o estado de um equipamento para o colocar à sua preferência, devido a uma falha de previsão do módulo. Estas falhas acontecem quando os valores de saída da rede neuronal associada a cada objecto não são os esperados. Há diferentes tipos de erros que provocam diferentes reacções do habitante:

- Equipamentos que têm de ser manualmente ajustados: resultam de uma falha no valor previsto pela rede. Este é o tipo de erro menos incomodativo;
- Equipamentos que estão desligados e deveriam estar ligados: nesta situação os equipamentos necessitam de intervenção inteiramente manual. Este é um tipo de erro pouco incomodativo;
- Equipamentos que estão ligados e deveriam estar desligados: esta representa a pior situação. Para além do consumo energético dispensável, o utilizador encara o sistema como intrusivo. É a principal razão que determina a lenta adopção deste tipo de sistemas nas nossas habitações.

Foi criado um conjunto de treino composto por actividades aleatórias em equipamentos também aleatórios. Este teste possibilita avaliar a velocidade de convergência das várias configurações e

os seus limites. Considera-se a necessidade de controlar manualmente o equipamento quando o valor da saída difere por um valor superior a 10% da gama de valores por ele aceite.

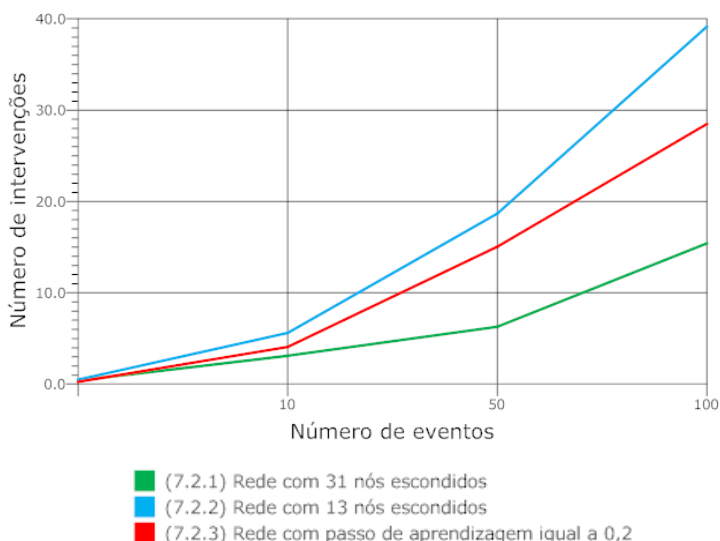


Figura 7.5 - Número de intervenções do habitante devido a um erro de previsão.

7.3 Análise de resultados

Como já era esperado os melhores resultados foram conseguidos com a configuração com 31 nós escondidos ($2 \times n^{\circ} \text{ de nós de entrada} + 1$). Esta é a estrutura que permite a aprendizagem de maior quantidade de informação. Com uma rede responsável por cada equipamento a configuração interna da rede rearranja-se de forma a “memorizar” as entradas, se forem treinadas o número de vezes suficiente. O grande inconveniente desta solução é o poder de processamento que exige - um teste completo para esta configuração demorava cerca de 25-35 minutos, dependendo de outros parâmetros da rede como o passo de aprendizagem ou o momento definidos.

A rede com 13 nós escondidos apresentou limitações em acompanhar com precisão o valor esperado. A menor capacidade em seguir transições bruscas era já esperada, pois uma rede com um número inferior de nós escondidos não possui a mesma pluralidade de soluções ao nível de reorganização interna.

A rede com 13 nós e com o valor do passo de aprendizagem duplicado e o valor do momento triplicado conseguiu encontrar uma organização interna da rede mais favorável a este teste, o que indica que conseguiu encontrar um máximo local mais desejável que o da rede de 13 nós escondidos sem estas alterações. É possível que esta alteração nos parâmetros tenha permitido encontrar o óptimo global desta configuração. É possível que com mais épocas de treino a

configuração inicial de 13 nós atingisse a mesma qualidade de resultados. Tal não aconteceria se, devido aos baixos valores de passo de aprendizagem e momento, a configuração ficasse presa num mínimo local do espaço de pesos.

Capítulo 8

Conclusões e Trabalho Futuro

Com o aparecimento de novas tecnologias, o desenvolvimento das já existentes e a contínua investigação no campo da inteligência artificial será de esperar que, certamente, um dia as soluções de domótica inteligente estarão suficientemente maduras para integrar as nossas habitações. A preocupação com a poupança de recursos é crescente, assim como a nossa dependência de equipamentos tecnologicamente avançados. Será concerteza uma questão de tempo até que encaremos a gestão da nossa habitação por uma máquina como uma necessidade.

8.1 Satisfação dos objectivos

A solução implementada permitiu a um utilizador recriar as actividades comuns numa habitação, que por sua vez as registava de forma a identificar e automatizar rotinas. A preocupação na redução do consumo energético foi também atendida, através da identificação dos períodos em que o sistema de aquecimento de água podia entrar em modo de poupança de energia e da prioridade em satisfazer a preferência de intensidade luminosa numa divisão usando ao máximo a luz natural. Assim, só no caso de a luz natural não ser suficiente para iluminação desejada é que as luzes artificiais são ligadas.

Ficou também provada a validade da arquitectura criada para a integração entre as várias aplicações.

8.2 Revisão do trabalho desenvolvido

Apesar dos resultados obtidos serem positivos, a verdade é que para se obter estes resultados foi necessário muito tempo de processamento computacional, aproximadamente 7 minutos para a solução com 13 nós e 28 minutos para a arquitectura com 31 nós. Estes valores estão aqui acentuados pelo carácter sucessivo do treino e das previsões, pois numa habitação as várias actividades estão suficientemente espaçadas temporalmente para que a rede neuronal processe os dados sem um atraso que comprometa o seu funcionamento, podendo se necessário acumular amostras para treinar em alturas de vazio. Comparando os resultados obtidos com os alcançados pelo projecto MavHome, conclui-se que as máquinas de vectores de suporte são mais indicadas para a previsão das actividades dos habitantes de uma habitação, o que era já esperado. A solução obtida com as MVS é sempre óptima e estas não necessitam de uma pré-configuração como as redes neuronais, em que é necessário definir o número de camadas e de nós antes de iniciar a fase de treino, o que torna difícil a adaptação de um rede a um cenário mutável como o de uma habitação.

8.3 Trabalho futuro

É possível evoluir a solução obtida através do uso de algoritmos aceleradores, que geralmente aumentam a velocidade de convergência da rede. Para tal era necessário alterar a plataforma utilizada. Com um período maior de testes seria também possível achar a configuração ideal da estrutura da rede e dos seus parâmetros.

A utilização de redes neuronais para o desenvolvimento de uma casa inteligente tem vantagens relativamente à utilização de outro tipo de algoritmos, sendo a maior o facto de uma rede neuronal não ser programada, mas possuir a capacidade de aprender e generalizar.

Uma solução a implementar deve utilizar vários destes algoritmos, aplicando-os nos cenários em que cada um apresenta os melhores resultados, para assim tirar partido das vantagens de cada um.

Referências

- [1] Amigo Project - <http://www.hitech-projects.com/euprojects/amigo/index.htm>. Fevereiro de 2008.
- [2] Neural Networks - Disponível em <http://www.herselfsai.com/2007/02/neural-networks.html>.
- [3] Neural Network Frequently Asked Questions - Disponível em <http://web.ift.uib.no/Fysisk/Teori/NEURO/neurons.html>
- [4] Neural Networks: Motivation, Theory and DANN - Disponível em <http://www.generation5.org/content/2000/nn00.asp>
- [5] Liang Haifeng. An Adaptive BP-Network Approach to Short Term Load Forecasting, 2004.
- [6] Multiple Layer Perceptron - Disponível em <http://ai.kaist.ac.kr/~jkim/cs570-2005/MLP1.ppt>, 2004.
- [7] Andrew W. Moore - Hidden Markov models - Disponível em <http://www.cs.cmu.edu/~awm>, 2003.
- [8] Magno Silva, Max Gerken - Um Algoritmo Acelerador Para Treinamento de Redes Neurais MLP.
- [9] Previsão de Séries de Tempo na Presença de Mudança Estrutural: Redes Neurais Artificiais e Modelos Estruturais - Disponível em <http://www.herselfsai.com/2007/03/backpropagation-networks.html>.
- [10] Leandro Loss - Redes Neurais. Disponível em <http://www.gsigma.ufsc.br/~loss>
- [11] Diane J. Cook - MavHome: An Agent-Based Smart Home. Disponível em <http://citeseer.ist.psu.edu/cook03mavhome.html>
- [12] MavHome: Managing an Adaptive Versatile Home. Disponível em <http://ailab.wsu.edu/mavhome/>
- [13] NeuronDotNet - Artificial Neural Networks in C#. Disponível em <http://neurondotnet.freehostia.com/>
- [14] Russell S., Norvig P. - Artificial intelligence: a modern approach, 2003
- [15] Roman Neruda - Kolmogorov Learning for FeedForward Networks.
- [16] Christopher M. Bishop - Neural Networks for Pattern Recognition, 1995.

[17]Fernando Augusto Carneiro Sereno - Regressão com Redes Neurais e Máquinas de Vectores de Suporte Aplicadas à Estimação do Peso Fetal, 2002.

[18]José Afonso - Predição de Séries no Tempo utilizando Redes Neurais, 1994.

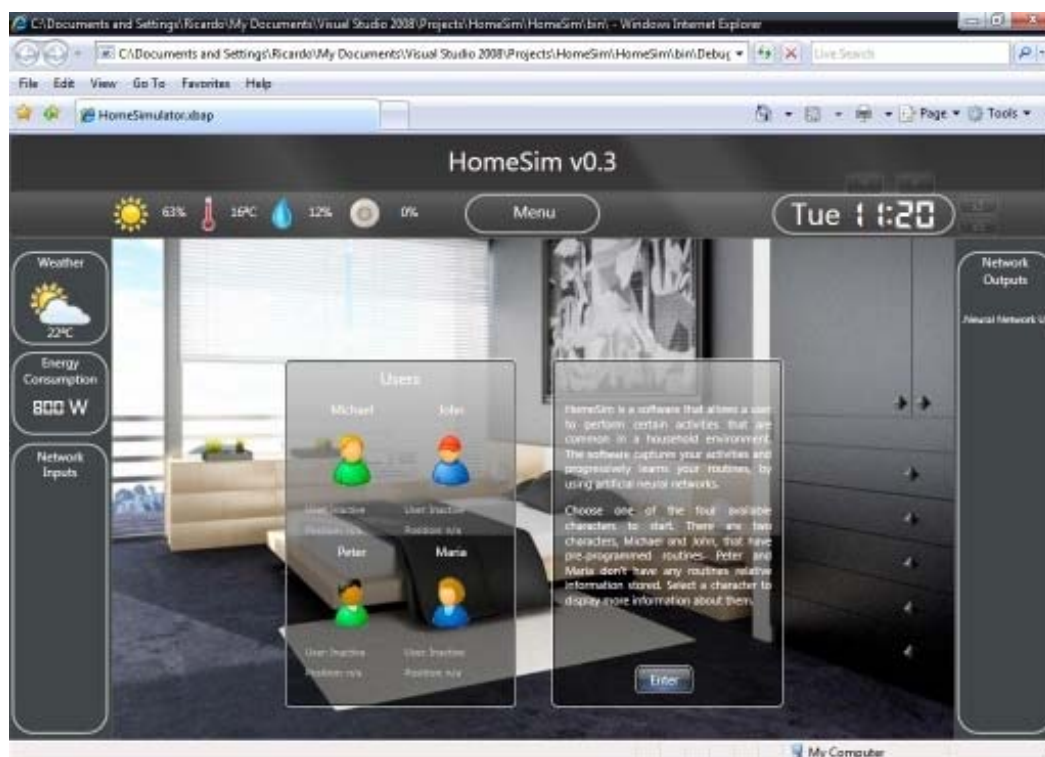
[19]MSDN: WF - Disponível em <http://msdn.microsoft.com/en-us/netframework/aa663322.aspx>.

[20]MSDN: WPF - Disponível em <http://msdn.microsoft.com/en-us/netframework/aa663326.aspx>.

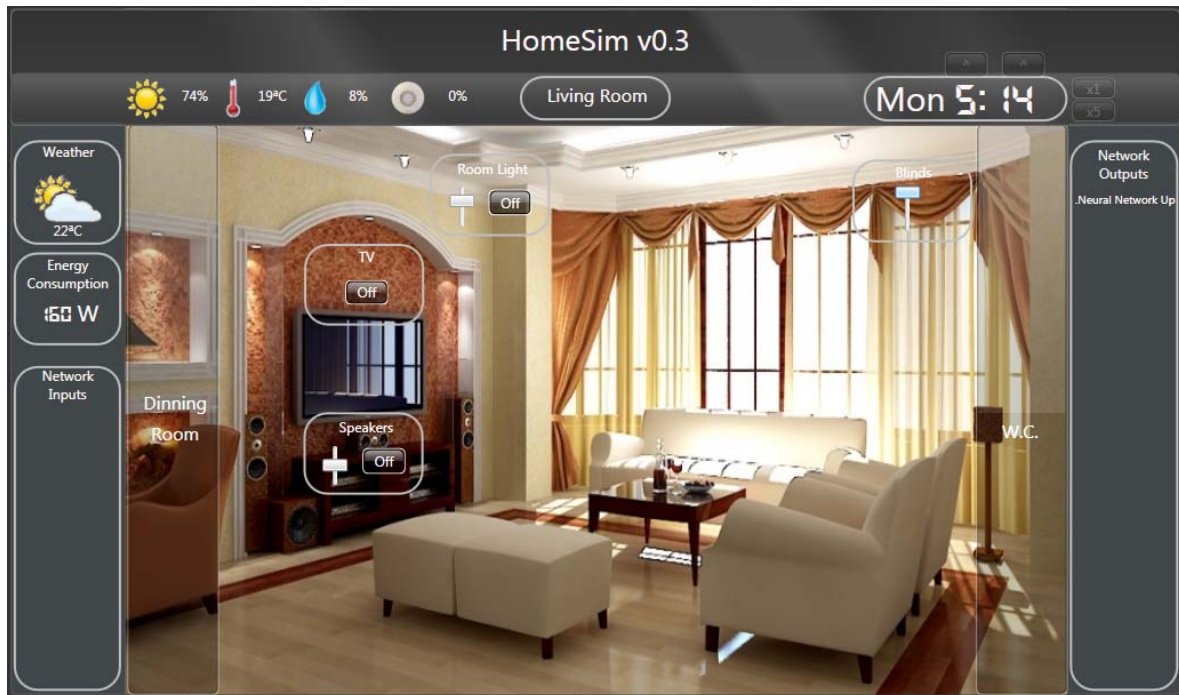
[21]MSDN: WCF - Disponível em <http://msdn.microsoft.com/en-us/netframework/aa663324.aspx>.

Anexos

Interface da aplicação HomeSim:



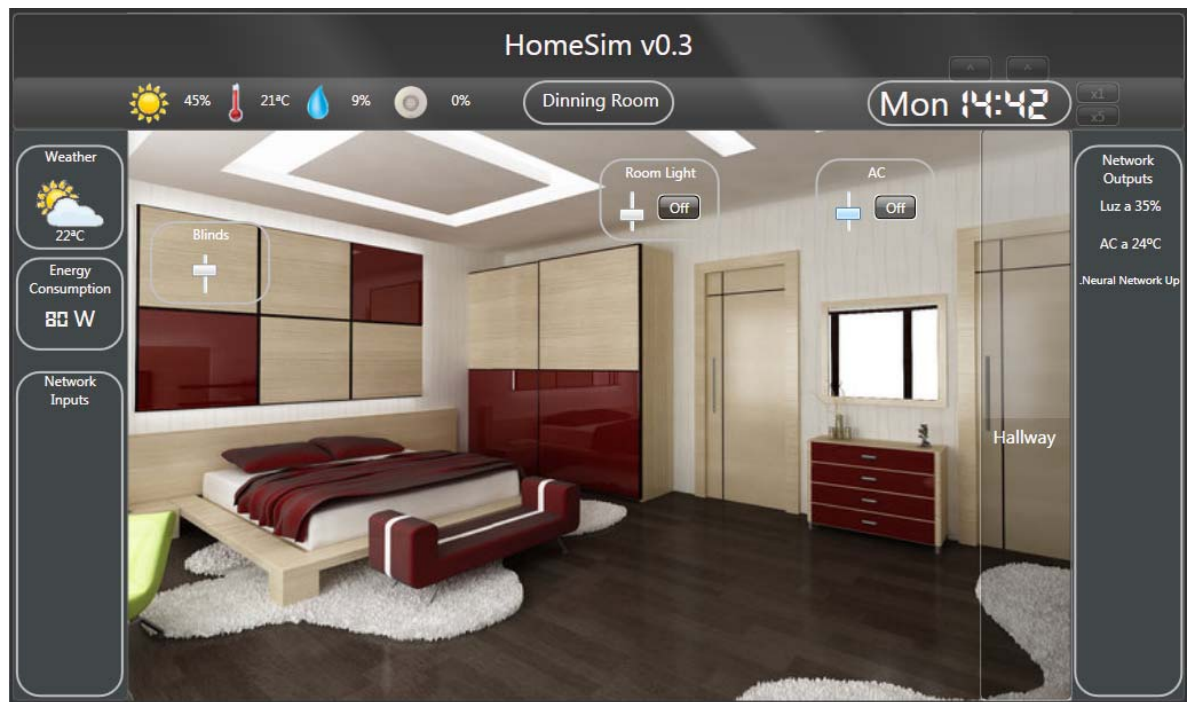
Anexos - Figura 1 - Menu de escolha de personagem.



Anexos - Figura 2 - Sala de estar.



Anexos - Figura 3 - Escritório.

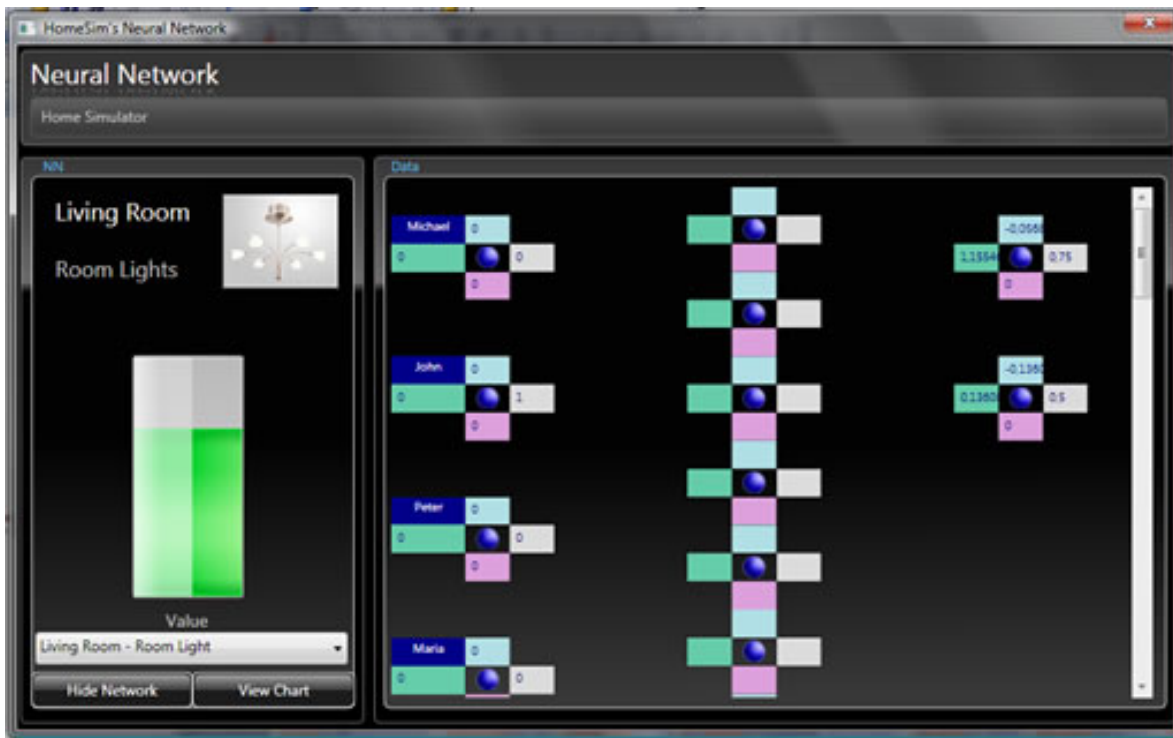


Anexos - Figura 4 - Quarto.



Anexos - Figura 5 - Quarto de banho.

Interface da aplicação Rede Neuronal Visual:



Anexos - Figura 6 - Quarto de banho.